ELSEVIER

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Journal of Complexity 19 (2003) 799–834

Journal of COMPLEXITY

http://www.elsevier.com/locate/jco

A recursive algorithm for the infinity-norm fixed point problem

Spencer Shellman and K. Sikorski*

School of Computing, University of Utah, Salt Lake City, UT 84112, USA

Received 18 September 2002; accepted 26 June 2003

Abstract

We present the PFix algorithm for the fixed point problem $f(x) = x$ on a nonempty domain $[a, b]$, where $d \geqslant 1$, $a, b \in \mathbb{R}^d$, and f is a Lipschitz continuous function with respect to the infinity norm, with constant $q \leqslant 1$. The computed approximation $\tilde{x}$ satisfies the residual criterion $||f(\tilde{x}) - \tilde{x}||_\infty \leqslant \varepsilon$, where $\varepsilon > 0$. In general, the algorithm requires no more than $\sum_{i=1}^{d} s^i$ function component evaluations, where $s \equiv \lceil \max(1, \log_2(||b-a||_\infty/\varepsilon)) \rceil + 1$. This upper bound has order $\mathcal{O}(\lceil \log_2^d(1/\varepsilon) \rceil)$ as $\varepsilon \to 0$. For the domain $[0,1]^d$ with $\varepsilon < 0.5$ we prove a stronger result, i.e., an upper bound on the number of function component evaluations is $\binom{d+r-1}{r-1} + 2\binom{d+r}{r+1}$, where $r \equiv \lceil \log_2(1/\varepsilon) \rceil$. This bound approaches $\mathcal{O}(r^d/d!)$ as $r \to \infty$ $(\varepsilon \to 0)$ and $\mathcal{O}(d^{r+1}/(r+1)!)$ as $d \to \infty$. We show that when $q < 1$ the algorithm can also compute an approximation $\tilde{x}$ satisfying the absolute criterion $||\tilde{x} - x^*||_\infty \leqslant \varepsilon$, where x^* is the unique fixed point of f. The complexity in this case resembles the complexity of the residual criterion problem, but with tolerance $\varepsilon(1-q)$ instead of ε. We show that when $q > 1$ the absolute criterion problem has infinite worst-case complexity when information consists of function evaluations. Finally, we report several numerical tests in which the actual number of evaluations is usually much smaller than the upper complexity bound.

Keywords: Fixed point; Bisection algorithm; Computational complexity; Lipschitz functional

1. Introduction

The development of constructive algorithms for approximating fixed points started in the 1920s with Banach's simple iteration algorithm [2]. Several algorithms

*Corresponding author.
E-mail address: sikorski@cs.utah.edu (K. Sikorski).

0885-064X/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jco.2003.06.001

have been developed since then, including homotopy continuation, simplicial and Newton-type methods [1,5,6,15,22]. It has been shown [10] that for Lipschitz functions with constant $q>1$ with respect to the infinity norm, the latter algorithms exhibit exponential complexity in the worst case (when computing ε-residual solutions $\tilde{x}: ||\tilde{x}-f(\tilde{x})||_\infty \leqslant \varepsilon$) and that the lower bound on the complexity is also exponential.

Fixed point problems with $q=1$ and $q<1$ but very close to one appear in many disciplines including economics, game theory (especially ergodic games), meromorphic functions, nonlinear differential equations and dynamical systems. In the study of dynamical systems with two degrees of freedom, such fixed point problems model conservative or dissipative systems depending on whether the mapping is area-preserving or area-contracting, respectively (see: [3,4,7–9,11,21]). In the theory of dynamical systems, the fixed points are called "periodic orbits" of such mappings [20]. Fixed point problems with $q<1$ are very common in numerical computation (for nonlinear problems and large scale linear systems).

Several algorithms for approximating a fixed point α of a Lipschitz function that is contractive ($q<1$) or nonexpanding ($q=1$) with respect to the second norm have been developed [12,18,19]. For $q<1$ and large dimension d, the Banach simple iteration algorithm $x_{i+1}=f(x_i)$ is optimal [14,18]. It requires $n=\lceil \log(1/\varepsilon)/\log(1/q)\rceil$ function evaluations to compute $\tilde{x}$ such that $||\tilde{x}-\alpha||_2 \leqslant \varepsilon||\alpha||_2$. This number of evaluations is very large when q is close to one. In the univariate case a class of bisection-envelope algorithms is optimal with respect to various error criteria [18]. For moderate dimension d and $q=1$, the interior ellipsoid algorithm is optimal [12,13]. This algorithm requires $c \cdot d \cdot \log(1/\varepsilon)$ function evaluations to compute an ε-residual approximation $\tilde{x}: ||\tilde{x}-f(\tilde{x})||_2 \leqslant \varepsilon$. We stress that the worst-case complexity of computing an ε-absolute approximation $\tilde{x}: ||\tilde{x}-\alpha||_2 \leqslant \varepsilon$ for $q=1$ is infinite [18]. This means that there exists no algorithm based on function evaluations that solves this problem for all functions in this class. For $q<1$ the interior ellipsoid algorithm [12] computes $\tilde{x}: ||\tilde{x}-\alpha|| \leqslant \varepsilon$ within $c \cdot d(\log(1/\varepsilon) + \log(1/(1-q)))$ function evaluations.

Our goal is to generalize the second norm results to the other practical case of the infinity norm. To this end we presented in [16,17] a BEDFix algorithm for the bivariate case. This algorithm computes an ε-residual solution for a bivariate function with $q \leqslant 1$ in no more than $2\lceil \log_2(1/\varepsilon)\rceil + 1$ function evaluations. We believe this to be an optimal lower bound on the worst-case complexity of the two-dimensional problem. It is known [10] that a simple uniform grid search method can compute an ε-residual solution for functions of dimension $d>2$ in no more than $\mathcal{O}(\varepsilon^{-d})$ function evaluations for any $q \geqslant 1$. In this paper we present the PFix algorithm, which computes ε-residual solutions for general $d \geqslant 1$ and $q=1$. This algorithm improves on the exponential worst-case complexity of the grid search method, and converges when $q=1$ unlike the simple iteration method. For an arbitrary domain $[a,b] \subseteq \mathbb{R}^d$, the algorithm requires no more than $\sum_{i=1}^{d} s^i$ $(= \mathcal{O}(\lceil \log_2^d(1/\varepsilon)\rceil)$ as $\varepsilon \to 0)$ function component evaluations, where $s \equiv \lceil \max(1, \log_2(||b-a||_\infty/\varepsilon))\rceil + 1$. On the domain $[0,1]^d$ with $\varepsilon<0.5$, an upper bound on the number of required function component evaluations is $\binom{d+r-1}{r-1} + 2\binom{d+r}{r+1}$, where $r = \lceil \log_2(1/\varepsilon) \rceil$; this bound is less than exponential in d and r. When $q<1$ the algorithm also computes ε-absolute solutions when executed with error tolerance $\varepsilon(1-q)$, and is more efficient than simple iteration when q is close to 1. When $q>1$ we show that the absolute criterion problem has infinite worst-case complexity when information consists of function evaluations. The PFix algorithm is recursive in that while it is solving for a d-dimensional function ($d>1$), it repeatedly calls itself on the function restricted to a $d-1$-dimensional subdomain. We stress that algorithms which are derived for the $q=1$ case, can also be much more efficient than the simple iteration for the case $q<1$, as demonstrated in our numerical tests. Such methods can be used to speed-up other iterative algorithms, like e.g. Newton's method, near the boundary of the region of convergence. In such cases Newton's method exhibits very slow linear convergence with contraction factor very close to one.

We do not know whether our derived upper bounds for PFix are sharp. We were not able to find testing functions that would result in the number of iterations close to the upper bound. By using the 2-dim BedFix algorithm in the interior loop we could improve the upper bound to be $O(\log(1/\varepsilon)^{d/2})$, as $\varepsilon \to 0$. We still do not know the optimal algorithm with a bound $cd\log(1/\varepsilon)$ with respect to the infinity norm.

We included in this paper a modification of the infinite complexity proof for the second norm and $q=1$ case, since we believed that it would also yield the same result for the infinity norm with $q=1$. Unfortunately, we were able to prove it only for the $q>1$ case. This could also be proven by using the second norm worst-case functions f_1 and f_2 [18] and modifying the adaptive generation of function values on page 23, to become $\beta * x + (1-\beta) * f_i(x)$, where $\beta = (q-1)/(\sqrt{(}d) - 1)$, $i = 1, 2$, and where $q>1$ is the Lipschitz constant in the infinity norm. We think that a modification to our proof technique will be able to show the same result for $q=1$.

2. Problem formulation

Given dimension $d \geqslant 1$ and vectors $a, b \in \mathbb{R}^d$ with $a_i \leqslant b_i$ for all $i \in I_d$, we define the domain

$$D_{a,b} \equiv [a_1, b_1] \times \cdots \times [a_d, b_d]$$

and the class $\mathfrak{F}_{a,b}$ of functions $f : D_{a,b} \to D_{a,b}$ that are Lipschitz continuous with constant 1 with respect to the infinity norm, i.e.,

$$\|f(x) - f(y)\| \leqslant \|x - y\| \quad \forall x, y \in D^d, \tag{1}$$

where $\|\cdot\| = \|\cdot\|_\infty$ henceforth. We define $D^d \equiv D_{0,(1,\dots,1)} = [0,1]^d$ and $\mathfrak{F}^d \equiv \mathfrak{F}_{0,(1,\dots,1)}$. We present an algorithm called PFix which, for every $f \in \mathfrak{F}_{a,b}$, computes a solution $\tilde{x} \equiv \tilde{x}(f) \in D_{a,b}$ that satisfies the residual criterion

$$||f(\tilde{x}) - \tilde{x}|| \leqslant \varepsilon \tag{2}$$

for a positive error tolerance ε.

We prove that for a general domain $[a, b]$, the number of required function component evaluations is bounded above by

$$n(d, s) \equiv \sum_{i=1}^{d} s^i = (s^d - 1)\frac{s}{s-1} = \mathcal{O}(\lceil \log_2^d(1/\varepsilon) \rceil) \quad \text{as } \varepsilon \to 0, \tag{3}$$

where $s \equiv \lceil \max(1, \log_2(||b-a||/\varepsilon)) \rceil + 1$. For the case $D_{a,b} = D^d = [0,1]^d$ and $0 < \varepsilon < 0.5$ (if $D_{a,b} = D^d$ and $\varepsilon \geqslant 0.5$ then $\tilde{x} = (0.5, \ldots, 0.5)$ satisfies (2)), we present a proof that the number of required function component evaluations is bounded above by

$$B(d, r) \equiv C(d, r) - C(d-1, r) + 2(C(d-1, r+2) - C(d-2, r+2)), \tag{4}$$

here $r \equiv \lceil \log_2(1/\varepsilon) \rceil$ and the function C is defined for integers $m \geqslant 1$ and d as

$$C(d, m) \equiv \begin{cases} 0, & d < 0, \\ \binom{d+m-1}{m-1}, & d \geqslant 0 \end{cases} \tag{5}$$

(where $\binom{n}{k} \equiv n!/(k!(n-k)!)$). To find an order bound for $B(d, r)$ we consider only the positive terms of (4), i.e., $B(d, r) \leqslant C(d, r) + 2C(d-1, r+2)$. As d is fixed and $r \to \infty$ ($\varepsilon \to 0$) the r^d term dominates in $\binom{d+r-1}{r-1} = \frac{r(r+1)\cdots(r+d-1)}{d!}$, so $C(d, r)$ becomes $\mathcal{O}(r^d/d!)$. As r is fixed and $d \to \infty$ the d^{r-1} term dominates in $\binom{d+r-1}{r-1} = \frac{(d+1)\cdots(d+r-1)}{(r-1)!}$, so $C(d, r)$ becomes $\mathcal{O}(d^{r-1}/(r-1)!)$. Similarly, as d is fixed and $r \to \infty$, $C(d-1, r+2)$ becomes $\mathcal{O}(r^{d-1}/(d-1)!)$, and as r is fixed and $d \to \infty$, $C(d-1, r+2)$ becomes $\mathcal{O}(d^{r+1}/(r+1)!)$. It follows that as d is fixed and $r \to \infty$, the bound $C(d, r) + 2C(d-1, r+2)$ becomes $\mathcal{O}(r^d/d!)$, and as r is fixed and $d \to \infty$, the bound becomes $\mathcal{O}(d^{r+1}/(r+1)!)$.

We show that for a function $f \in \mathfrak{F}_{a,b}$ that is Lipschitz continuous with constant q ($0 < q < 1$), i.e.,

$$||f(x) - f(y)|| \leqslant q||x - y|| \quad \forall x, y \in D^d, \tag{6}$$

the algorithm computes a solution $\tilde{x} \in D^d$ satisfying the absolute criterion

$$||\tilde{x} - x^*|| \leqslant \varepsilon, \tag{7}$$

where x^* is a fixed point of f, when executed with error tolerance $\varepsilon(1-q)$. In the absolute criterion case with $[a, b] = [0, 1]^d$ and $0 < \varepsilon < 0.5$, every $f \in \mathfrak{F}^d$ requires no more than $B(d, r_q)$ function component evaluations, where $r_q \equiv \lceil \log_2(1/\varepsilon) + \log_2(1/(1-q)) \rceil$. In addition we prove that for the class of functions that are Lipschitz continuous with constant $q > 1$, the absolute-criterion fixed point problem has infinite worst-case complexity when information consists of function evaluations. Finally, we report several numerical tests in which the actual number of evaluations is usually several orders of magnitude less than the given upper bound.

Throughout this paper, when we refer to a function as Lipschitz continuous we mean that it is Lipschitz continuous with constant 1, unless specified otherwise.

3. Definitions

We define an *index set* as an ordered set of positive integers $\{s_1, \ldots, s_k\}$ such that $s_1 < \cdots < s_k$, and define the index set $I_d \equiv \{1, \ldots, d\}$ for dimension $d \geqslant 1$. We assume that subsets, intersections, and unions of index sets are also index sets. When we take a subset $S \subseteq I_d$ we denote $I_d - S$ by $\bar{S}$.

For all $i \in I_d$ we define the ith unit vector $e^i \in \mathbb{R}^d$ such that $e^i_i = 1$ and $e^i_j = 0$ for all $j \in I_d - \{i\}$. Given two vectors $u, v \in \mathbb{R}^d$, we say that $u \leqslant v$ if and only if $u_i \leqslant v_i$ for $i = 1, \ldots, d$, and define $u \geqslant v$, $u < v$, and $u > v$ similarly. Hence $D_{u,v}$ is the set of all vectors $x \in \mathbb{R}^d$ such that $u \leqslant x \leqslant v$. Given $x \in \mathbb{R}^d$, $i \in I_d$, and $c \in \mathbb{R}$, we define

$$x[i \leftarrow c] \equiv x + (c - x_i)e^i.$$

Given a nonempty index set $S \equiv \{s_1, \ldots, s_k\} \subset I_d$, we define the domain

$$D^S_{a,b} \equiv D_{(a_{s_1}, \ldots, a_{s_k}),(b_{s_1}, \ldots, b_{s_k})}$$

and the operators $\Lambda_S : \mathbb{R}^d \to \mathbb{R}^k$, $\Phi_S : \mathbb{R}^k \to 2^{\mathbb{R}^d}$, and $\Psi_S : \mathbb{R}^k \times \mathbb{R}^{d-k} \to \mathbb{R}^d$ as

$$\Lambda_S(x) \equiv (x_{s_1}, \ldots, x_{s_k}),$$

$$\Phi_S(x) \equiv \{y \in \mathbb{R}^d\colon\ \Lambda_S(y) = x\}$$

and

$$\Psi_S(x, y) \equiv z \in \mathbb{R}^d \text{ s.t. } \Lambda_S(z) = x \text{ and } \Lambda_{\bar{S}}(z) = y.$$

Given $\varepsilon > 0$ we define the extended range

$$D^\varepsilon_{a,b} \equiv D_{(a_1 - \varepsilon, \ldots, a_d - \varepsilon),(b_1 + \varepsilon, \ldots, b_d + \varepsilon)}$$

and the extended class $\mathfrak{F}^\varepsilon_{a,b}$ of Lipschitz continuous functions having domain $D_{a,b}$ and range $D^\varepsilon_{a,b}$ (so that $\mathfrak{F}_{a,b} \subseteq \mathfrak{F}^\varepsilon_{a,b}$). We will show later that every function in $\mathfrak{F}^\varepsilon_{a,b}$ has an ε-residual solution whether or not it has a fixed point.

Given f in $\mathfrak{F}^\varepsilon_{a,b}$, we define sets of componentwise ε-residual solutions for $i = 1, \ldots, d$,

$$R^\varepsilon_{a,b,i}(f) \equiv \{x \in D_{a,b}\colon\ |f_i(x) - x_i| \leqslant \varepsilon\}$$

and the set of residual solutions

$$R^\varepsilon_{a,b}(f) \equiv \bigcap_{i=1}^{d} R^\varepsilon_{a,b,i}(f).$$

Given a nonempty index set $S \equiv \{s_1, \dots, s_k\} \subseteq I_d$, we define

$$R^{\varepsilon,S}_{a,b}(f) \equiv \bigcap_{i=1,\dots,k} R^{\varepsilon}_{a,b,s_i}(f).$$

4. Supporting theory

In this section we present theorems that will be used in the development of the algorithm.

4.1. Existence theorem

As we mentioned in Section 3, one of our goals is to show that a residual solution exists for every function in $\mathfrak{F}^{\varepsilon}_{a,b}$. The following is a nonconstructive proof of this fact; the algorithm analysis in Section 5.2 will provide a constructive proof.

Theorem 4.1. *Suppose we have $d \geqslant 1$, $a, b \in \mathbb{R}^d$, and $f \in \mathfrak{F}^{\varepsilon}_{a,b}$. Then $R^{\varepsilon}_{a,b}(f) \neq \emptyset$.*

Proof. We define the projection $P : D^{\varepsilon}_{a,b} \to D_{a,b}$ as

$$P(x) = (\max(a_1, \min(b_1, x_1)), \dots, \max(a_d, \min(b_d, x_d))) \quad \forall x \in D^{\varepsilon}_{a,b}$$

and the function $f' : D^{\varepsilon}_{a,b} \to D^{\varepsilon}_{a,b}$ as $f'(\cdot) \equiv f(P(\cdot))$. Clearly f' is continuous, so by the Brouwer theorem it has a fixed point x^*. We now set $y^* = P(x^*)$ and obtain

$$\begin{aligned} ||f(y^*) - y^*|| &= ||f(y^*) - f'(x^*) + x^* - y^*|| = ||f'(x^*) - f'(x^*) + x^* - y^*|| \\ &= ||x^* - y^*|| \leqslant \varepsilon \end{aligned}$$

so that $y^* \in R^{\varepsilon}_{a,b}(f)$. $\square$

4.2. Bisection theorem

We present a theorem that determines the locations of ε-residual solutions in the univariate case. The theorem forms the basis for a method to compute a residual solution for a function in $\mathfrak{F}^{\varepsilon}_{a,b}$, similar to the bisection method for approximating a fixed point of a function in $\mathfrak{F}_{a,b}$ [18]. It is based on the observation that $|f(x) - x^*| \leqslant |x - x^*|$ for any fixed point x^* of f and x in the domain of f.

Theorem 4.2. *The following hold for every $f \in \mathfrak{F}^{\varepsilon}_{a,b}$ with $a, b \in \mathbb{R}$:*

(i) $R^{\varepsilon}_{a,b}(f) \neq \emptyset$.
(ii) *Take $c \in [a, b]$. If $f(c) > c$ then $f(c^+) \geqslant c^+$ and $R^{\varepsilon}_{a,b}(f) \cap [c^+, b] \neq \emptyset$, where $c^+ \equiv \min(b, (f(c) + c)/2)$. If $f(c) < c$ then $f(c^-) \leqslant c^-$ and $R^{\varepsilon}_{a,b}(f) \cap [a, c^-] \neq \emptyset$, where $c^- \equiv \max(a, (f(c) + c)/2)$.*

(iii) *If* $c \in [a, a+\varepsilon/2]$ *and* $f(c) \leqslant c$ *then* $a \in R^{\varepsilon}_{a,b}(f)$. *If* $c \in [b-\varepsilon/2, b]$ *and* $f(c) \geqslant c$ *then* $b \in R^{\varepsilon}_{a,b}(f)$.

(iv) *If* $c \in [a, a+\varepsilon]$ *and* $f(c) \leqslant c - \varepsilon$ *then* $a \in R^{\varepsilon}_{a,b}(f)$. *If* $c \in [b-\varepsilon, b]$ *and* $f(c) \geqslant c + \varepsilon$ *then* $b \in R^{\varepsilon}_{a,b}(f)$.

(v) *If* $f(c) = c$ *for some* $c \in [a,b]$, *then* $c' \in R^{\varepsilon}_{a,b}(f)$ *for all* $c' \in [a,b]$ *such that* $|c' - c| \leqslant \varepsilon/2$.

Proof. We obtain (i) as a result of Theorem 4.1.

We prove the first case of (ii); the second case is analogous. The Lipschitz continuity of f yields $|f(c^+) - f(c)| \leqslant c^+ - c \leqslant (f(c) - c)/2$, so

$$f(c^+) \geqslant f(c) - \tfrac{1}{2}(f(c) - c) = c + \tfrac{1}{2}(f(c) - c) \geqslant c^+.$$

Suppose that $c^+ \notin R^{\varepsilon}_{a,b}(f)$ (otherwise (ii) obviously holds). We have shown that $f(c^+) \geqslant c^+$, so we must have $f(c^+) > c^+ + \varepsilon$. If there exists $c^* \in R^{\varepsilon}_{a,b}(f) \cap [a, c^+)$ then $f(c^*) \leqslant c^* + \varepsilon$ and $f(c^+) - f(c^*) > c^+ - c^*$, a contradiction. It follows from (i) that $R^{\varepsilon}_{a,b}(f) \cap (c^+, b] \neq \emptyset$.

We prove the first case (iii); the second case is analogous. The Lipschitz continuity of f yields $|f(c) - f(a)| \leqslant c - a \leqslant \varepsilon/2$, so

$$f(a) \leqslant f(c) + \tfrac{1}{2}\varepsilon \leqslant c + \tfrac{1}{2}\varepsilon \leqslant a + \varepsilon.$$

Since $f(a) \geqslant a - \varepsilon$, we have $a \in R^{\varepsilon}_{a,b}(f)$.

We prove the first case (iv); the second case is analogous. We define $c' \equiv \max(a, c - \varepsilon/2)$. The Lipschitz continuity of f yields $|f(c) - f(c')| \leqslant c - c' \leqslant \varepsilon/2$, so

$$f(c') \leqslant f(c) + \tfrac{1}{2}\varepsilon \leqslant c - \tfrac{1}{2}\varepsilon \leqslant c'.$$

We now apply (iii) to c'.

To prove (v) we suppose that $f(c) = c$ and take $c' \in [a,b]$ such that $|c' - c| \leqslant \varepsilon/2$. We obtain

$$|f(c') - c'| = |f(c') - c + c - c'| \leqslant |f(c') - f(c)| + |c' - c| \leqslant 2|c' - c| \leqslant \varepsilon. \qquad \square$$

The following lemma allows us to apply Theorem 4.2 to a multivariate function defined on a subset of its domain as if it were a univariate function.

Lemma 4.3. *Suppose we have* $f \in \mathfrak{F}^{\varepsilon}_{a,b}$ *with* $a, b \in \mathbb{R}^d$ $(d > 1)$, $i \in I_d$, *and a subset* $L \subseteq D_{a,b}$ *such that*

- *for every* $c \in [a_i, b_i]$ *there exists a unique* $x \equiv x(c) \in L$ *such that* $x_i = c$, *and*
- *for all* $u, v \in L$, $||u - v|| = |u_i - v_i|$.

If we define a function f' *on* $[a_i, b_i]$ *as* $f'(\cdot) \equiv f_i(x(\cdot))$, *then* $f' \in \mathfrak{F}^{\varepsilon}_{a_i,b_i}$.

Proof. Since f_i has range $[a_i - \varepsilon, b_i + \varepsilon]$, so does f'. We show that f' is Lipschitz continuous. For all $c_1, c_2 \in [a_i, b_i]$,

$$\begin{aligned} \|f'(c_1) - f'(c_2)\| &= \|f_i(x(c_1)) - f_i(x(c_2))\| \leqslant \|x(c_1) - x(c_2)\| \\ &= |x_i(c_1) - x_i(c_2)| = |c_1 - c_2|. \qquad \square \end{aligned}$$

4.3. Recursive domain theorem

The basic operation of the algorithm is to follow a path of points that satisfy the residual criterion in components 1 through k ($k < d$), in search of a point which also satisfies the residual criterion in component $k+1$. Given $f \in \mathfrak{F}^{\varepsilon}_{a,b}$, $x \in R^{\varepsilon,\{1,\dots,k\}}_{a,b}(f)$, and $h \in \mathbb{R} - \{0\}$, the next point in the path (call it z) must satisfy $z \in R^{\varepsilon,\{1,\dots,k\}}_{a,b}(f)$ and $z_{k+1} = x_{k+1} + h$. Assuming that $h > 0$ (the case $h < 0$ is similar), we define P as the set of all points $u \in \mathbb{R}^d$ such that $\Lambda_{\{k+2,\dots,d\}}(u) = \Lambda_{\{k+2,\dots,d\}}(x)$ and $\|u - x\| = u_{k+1} - x_{k+1} \leqslant h$. We see that P is a $k+1$-dimensional pyramid with height h and apex x, and we define g as the function f limited to the intersection of the domain of f with the base of P. Fig. 1 illustrates a three-dimensional pyramid in 3-space trimmed by the unit cube. The following theorem can be applied to this situation by setting $S = \{1, \dots, k\}$. It shows that g is in a function class which enables an algorithm execution on g to obtain a point satisfying the requirements for z.

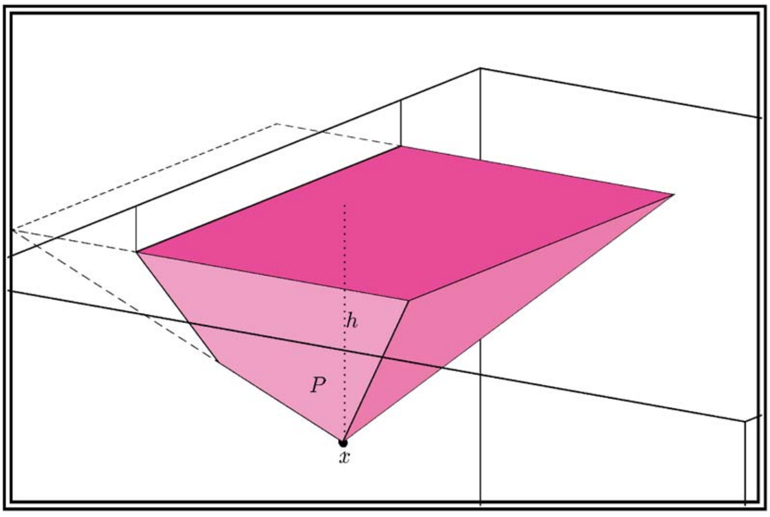

Fig. 1. A three-dimensional pyramid clipped by the unit cube in 3-space (see Theorem 4.4).

Theorem 4.4. *Suppose we have* $f \in \mathfrak{F}^{\varepsilon}_{a,b}$ *with* $a, b \in \mathbb{R}^d$ *and* $d > 1$, *an index set* $S \equiv \{s_1, \dots, s_k\} \subseteq I_d$ *with* $1 \leqslant k < d$, *and points* $x \in R^{\varepsilon,S}_{a,b}(f)$ *and* $y \in D^{\bar{S}}_{a,b}$. *We define* $a', b' \in \mathbb{R}^k$ *as*

$$a'_i \equiv \max(a_{s_i}, x_{s_i} - ||y - \Lambda_{\bar{S}}(x)||), \quad b'_i \equiv \min(b_{s_i}, x_{s_i} + ||y - \Lambda_{\bar{S}}(x)||),$$
$$i = 1, \dots, k.$$

Then the function $g(\cdot) \equiv \Lambda_S(f(\Psi_S(\cdot, y)))$ *with domain* $D_{a',b'}$ *is in* $\mathfrak{F}^{\varepsilon}_{a',b'}$, $R^{\varepsilon}_{a',b'}(g)$ *is nonempty, and for every* $x^* \in R^{\varepsilon}_{a',b'}(g)$, *we have* $\Psi_S(x^*, y) \in R^{\varepsilon,S}_{a,b}(f)$ *and* $||\Psi_S(x^*, y) - x|| = ||y - \Lambda_{\bar{S}}(x)||$.

Proof. To show that $g \in \mathfrak{F}^{\varepsilon}_{a',b'}$ we must first show that for all $w \in D_{a',b'}$, $\Lambda_S(f(\Psi_S(w, y))) \in D^{\varepsilon}_{a',b'}$. We choose $w \in D_{a',b'}$ and observe that by the definitions of a' and b', $||w - \Lambda_S(x)|| \leqslant ||y - \Lambda_{\bar{S}}(x)||$. We choose $i \in S$; since $f \in \mathfrak{F}^{\varepsilon}_{a,b}$, we must have $f_i(\Psi_S(w, y)) \in [a_i - \varepsilon, b_i + \varepsilon]$. We assume for sake of contradiction that either $f_i(\Psi_S(w, y)) < x_i - ||y - \Lambda_{\bar{S}}(x)|| - \varepsilon$ or $f_i(\Psi_S(w, y)) > x_i + ||y - \Lambda_{\bar{S}}(x)|| + \varepsilon$. Then

$$\begin{aligned} ||f(\Psi_S(w, y)) - f(x)|| &\geqslant |f_i(\Psi_S(w, y)) - f_i(x)| \\ &= |f_i(\Psi_S(w, y)) - x_i + (x_i - f_i(x))| \\ &\geqslant |f_i(\Psi_S(w, y)) - x_i| - \varepsilon > ||y - \Lambda_{\bar{S}}(x)|| \\ &= \max(||y - \Lambda_{\bar{S}}(x)||, ||w - \Lambda_S(x)||) = ||\Psi_S(w, y) - x||, \end{aligned}$$

which contradicts the Lipschitz continuity of f. Hence $\Lambda_S(f(\Psi_S(w, y))) \in D^{\varepsilon}_{a',b'}$. The Lipschitz continuity of g follows trivially from the Lipschitz continuity of f, so $g \in \mathfrak{F}^{\varepsilon}_{a',b'}$. The nonemptiness of $R^{\varepsilon}_{a',b'}(g)$ is a result of Theorem 4.1.

For arbitrary $x^* \in R^{\varepsilon}_{a',b'}(g)$ we define $v \equiv \Psi_S(x^*, y)$. Since $||\Lambda_S(f(v)) - \Lambda_S(v)|| = ||g(x^*) - x^*|| \leqslant \varepsilon$, we obtain $v \in R^{\varepsilon,S}_{a,b}(f)$. Finally, since $x^* \in D_{a',b'}$, we obtain $||\Lambda_S(v) - \Lambda_S(x)|| = ||x^* - \Lambda_S(x)|| \leqslant ||y - \Lambda_{\bar{S}}(x)||$, so that $||v - x|| = ||y - \Lambda_{\bar{S}}(x)||$. □

5. The PFix algorithm

In this section we describe the PFix algorithm and determine its complexity.

5.1. Algorithm description

We list below the steps that the algorithm follows to compute $\tilde{x}$ satisfying $||f(\tilde{x}) - \tilde{x}|| \leqslant \varepsilon$, i.e., $\tilde{x} \in R^{\varepsilon}_{a,b}(f)$, where $a, b \in \mathbb{R}^d$, $d > 0$, and $f \in \mathfrak{F}^{\varepsilon}_{a,b}$. We assume $f \in \mathfrak{F}^{\varepsilon}_{a,b}$ instead of $f \in \mathfrak{F}_{a,b}$ because the algorithm may call itself recursively on functions in the larger class.

Steps 2–6 constitute a univariate bisection loop. When $d = 1$ this loop locates $\tilde{x}$ satisfying $|f(\tilde{x}) - \tilde{x}| \leqslant \varepsilon$; when $d > 1$ it is executed by recursive calls of depth $d - 1$.

1. If $d > 1$ then go to step 7, otherwise ($d = 1$) set $x^- := a$, and $x^+ := b$. If $x^- = x^+$ then terminate with $\tilde{x} := x^-$.
2. Set $x := (x^+ + x^-)/2$ and evaluate $u := f(x)$. If $|u - x| \leqslant \varepsilon$ then terminate with $\tilde{x} := x$.
3. If $x^- = a$, $x - x^- \leqslant \varepsilon$, and $u < x - \varepsilon$ then terminate with $\tilde{x} := x^-$. If $x^+ = b$, $x^+ - x \leqslant \varepsilon$, and $u > x + \varepsilon$ then terminate with $\tilde{x} := x^+$.
4. If $u > x + \varepsilon$ then set $x^- := \min(x^+, (x + u)/2)$, otherwise ($u < x - \varepsilon$) set $x^+ := \max(x^-, (x + u)/2)$. If $x^- = x^+$ then terminate with $\tilde{x} := x^-$.
5. If $x^- = a$, $x^+ - x^- \leqslant \varepsilon/2$, and $u < x - \varepsilon$ then terminate with $\tilde{x} := a$. If $x^+ = b$, $x^+ - x^- \leqslant \varepsilon/2$, and $u > x + \varepsilon$ then terminate with $\tilde{x} := b$.
6. If $x^- \neq a$, $x^+ \neq b$, and $x^+ - x^- \leqslant \varepsilon$ then terminate with $\tilde{x} := (x^+ + x^-)/2$, otherwise go to step 2.

   Steps 9–13 constitute a multivariate bisection loop which is executed when $d > 1$. This loop ensures that every evaluation point x satisfies $|f_i(x) - x_i| \leqslant \varepsilon$ for $i = 1, \ldots, d - 1$, by recursively invoking the algorithm on f restricted to a $d - 1$-dimensional subset of $D_{a,b}$. When the distance between successive evaluation points becomes sufficiently small, the loop is able to locate a point $\tilde{x}$ satisfying $||f(\tilde{x}) - \tilde{x}|| \leqslant \varepsilon$.
7. If $||b - a|| \leqslant 2\varepsilon$ then set $x := ((a_1 + b_1)/2, \ldots, (a_d + b_d)/2)$, then terminate with the following value of $\tilde{x}$:

$$\tilde{x}_i := \begin{cases} a_i, & a_i = b_i, \\ \max(a_i, \min(b_i, f_i(x))), & a_i < b_i, \end{cases} \quad i = 1, \ldots, d.$$

8. Set $S := \{1, \ldots, d - 1\}$. Recursively execute the algorithm on the function g defined on the domain $D_{\Lambda_S(a), \Lambda_S(b)}$ as

$$g(\cdot) \equiv \Lambda_S\left(f\left(\Psi_S\left(\cdot, \frac{a_d + b_d}{2} \right) \right) \right),$$

   to obtain $x^* \in R^{\varepsilon}_{\Lambda_S(a), \Lambda_S(b)}(g)$. Set $x := \Psi_S(x^*, (a_d + b_d)/2)$. Set $x^- := x[d \leftarrow a_d]$ and $x^+ := x[d \leftarrow b_d]$. If $x_d^- = x_d^+$ then terminate with $\tilde{x} := x$.
9. Evaluate $u_d := f_d(x)$. If $|u_d - x_d| \leqslant \varepsilon$ then terminate with $\tilde{x} := x$.
10. If $u_d > x_d + \varepsilon$ then set $x^- := x$, otherwise ($u_d < x_d - \varepsilon$) set $x^+ := x$.
11. If $x_d^- = a_d$ (resp. $x_d^+ = b_d$) and $x_d^+ - x_d^- \leqslant \varepsilon$ then do the following: Set $y := a_d$ (resp. $y := b_d$). Theorem 4.4 applied to f, S, x^+ (resp. x^-), and y yields definitions of a', b', and $g \in \mathfrak{F}^{\varepsilon}_{a',b'}$. Recursively execute the algorithm on g to obtain $x^* \in R^{\varepsilon}_{a',b'}(g)$. Terminate with $\tilde{x} := \Psi_S(x^*, y)$.
12. If $x_d^- = a_d$ (resp. $x_d^+ = b_d$) then do the following: Set $y := (x_d^+ + x_d^-)/2$. Theorem 4.4 applied to f, S, x^+ (resp. x^-), and y yields definitions of a', b', and $g \in \mathfrak{F}^{\varepsilon}_{a',b'}$.

Recursively execute the algorithm on g to obtain $x^* \in R^{\varepsilon}_{a',b'}(g)$. Set $x := \Psi_S(x^*, y)$. Go to step 9.

13. Set $y := (x_d^+ + x_d^-)/2$. Apply Theorem 4.4 to f, S, x^-, and y to obtain definitions $a^- \equiv a'$, $b^- \equiv b'$, and $g^- \equiv g \in \mathfrak{F}^{\varepsilon}_{a',b'}$. Apply Theorem 4.4 to f, S, x^+, and y to obtain definitions $a^+ \equiv a'$, $b^+ \equiv b'$, and $g^+ \equiv g \in \mathfrak{F}^{\varepsilon}_{a',b'}$. Define $a^*, b^* \in D_{a,b}$ such that for $i = 1, \ldots, d-1$, $a_i^* = \max(a_i^+, a_i^-)$ and $b_i^* = \min(b_i^+, b_i^-)$. Define the function $g \equiv g^+ = g^-$ restricted to the domain D_{a^*,b^*}. Recursively execute the algorithm on g to obtain $x^* \in R^{\varepsilon}_{a^*,b^*}(g)$. Set $x := \Psi_S(x^*, y)$. If $x_d^+ - x_d^- \leqslant 2\varepsilon$ then terminate with $\tilde{x} := x$, otherwise go to step 9.

Fig. 2 illustrates the PFix algorithm when $d = 2$. The dots on the dashed center line of the unit square represent the evaluation points of the univariate bisection loop, which is called recursively before the top-level multivariate bisection loop. Once the algorithm finds a point x^* that is a residual solution in the first component along this line, it searches the base of a triangle whose apex is x^* to find another such point. (We note that the part of the base that lies outside of the unit square does not need to be considered, hence the clipping of the triangle in Fig. 2.) The algorithm proceeds in this manner, searching the bases of ever smaller triangles until it finds a point that is a residual solution in both components.

Fig. 3 illustrates the PFix algorithm when $d = 3$. A dashed line surrounds a center plane of the unit cube; the algorithm executes a recursive call on this plane to obtain a point x^* that is a residual solution in the first two components. (For this reason the triangles in the center plane resemble those in Fig. 2.) Having done this, the algorithm searches the base of a pyramid whose apex is x^* to find another such point. (The part of the base that lies outside of the unit cube does not need to be

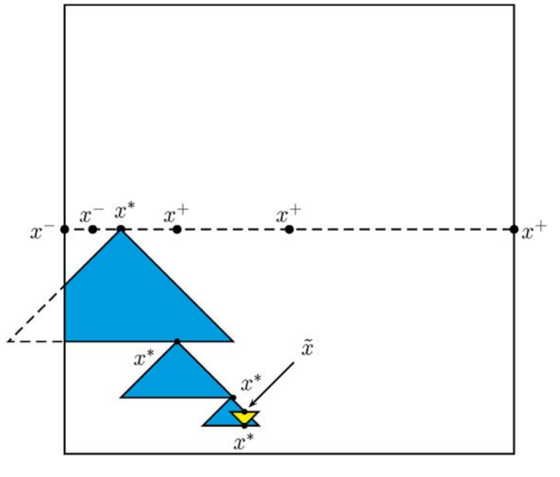

Fig. 2. Illustration of the algorithm when $d = 2$.

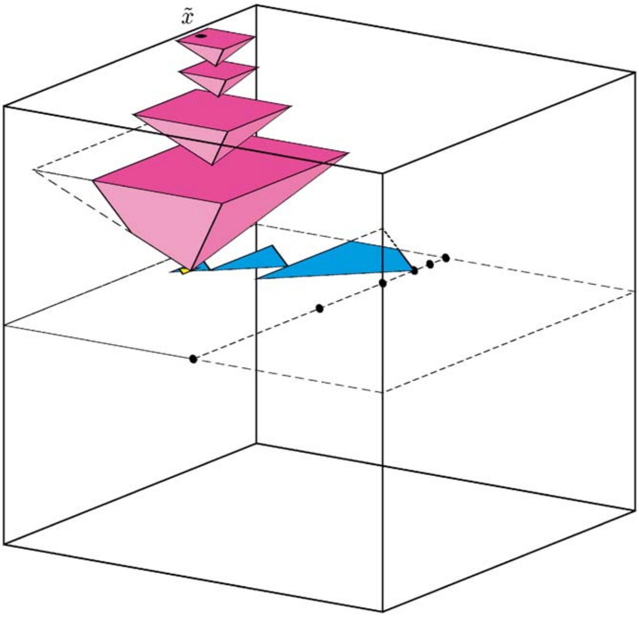

Fig. 3. Illustration of the algorithm when $d = 3$.

considered.) The algorithm searches the bases of ever smaller pyramids until it finds a point that is a residual solution in all three components.

In more than three dimensions, PFix searches the bases of successively smaller hyperpyramids to find residual points. The analysis of Section 5.2 shows how PFix limits the part of a base that needs to be searched for a residual solution point, in such a way that the residual solution points converge to a point that is also a residual solution in the next component.

5.2. Convergence

The following theorem includes a constructive proof of Theorem 4.1. In addition to proving that a residual solution exists, it shows that the PFix algorithm as described in Section 5.1 converges to such a solution while satisfying the complexity bound (3).

Theorem 5.1. *Suppose we have* $d \geqslant 1$, $a, b \in \mathbb{R}^d$, *and* $f \in \mathfrak{F}^{\varepsilon}_{a,b}$. *Then* $R^{\varepsilon}_{a,b}(f) \neq \emptyset$. *Furthermore, the PFix algorithm locates a point in* $R^{\varepsilon}_{a,b}(f)$ *using no more than* $n(d,s) \equiv \sum_{i=1}^{d} s^i$ *function component evaluations, where* $s \equiv \lceil \max(1, \log_2(||b-a||/\varepsilon)) \rceil + 1$.

Proof. We first prove the hypothesis in the case $d = 1$, then show that if $d > 1$ and the hypothesis is true for dimension $d - 1$, then it is also true for dimension d. We introduce a variable p, which counts the number of function component evaluations for purposes of complexity analysis.

The case $d = 1$: We first consider the case $d = 1$, and show that the algorithm returns $\tilde{x}$ satisfying the statement

(i) $|f(\tilde{x}) - \tilde{x}| \leqslant \varepsilon$,

using no more than s function component evaluations.

Step 1: We set $p \coloneqq 0$ at the beginning of step 1.

The following statements are true at the end of step 1:

(ii) $x^- < x^+$. This is due to the fact that $x^- = a \leqslant b = x^+$, and the decision to terminate if $x^- = x^+$ (in which case Theorem 4.2(i)) shows that $\tilde{x} = x^-$ satisfies statement (i).

(iii) $x^+ - x^- \leqslant 2^{-p}(b - a)$. This follows from setting $p \coloneqq 0$.

(iv) $R^{\varepsilon}_{a,b}(f) \cap [x^-, x^+] \neq \emptyset$. This is a result of Theorem 4.2(i).

(v) If $x^- \neq a$ then $f(x^-) \geqslant x^-$. If $x^+ \neq b$ then $f(x^+) \leqslant x^+$. These hold because $x^- = a$ and $x^+ = b$.

We will prove in the analysis of step 4 that statements (ii)–(v) are true at the beginning of each execution of step 2.

Univariate bisection loop

Steps 2–6 constitute the univariate bisection loop.

Step 2: We set $p \coloneqq p + 1$ at the beginning of step 2.

Step 2 sets $x \coloneqq (x^+ + x^-)/2$ and evaluates $u \coloneqq f(x)$. If $|u - x| \leqslant \varepsilon$ then $\tilde{x} \coloneqq x$ satisfies statement (i) and the algorithm terminates. Otherwise the following statement holds:

(vi) $|u - x| > \varepsilon$.

Step 3: In step 3, Theorem 4.2(iv) implies the following: If $x^- = a$, $x - x^- \leqslant \varepsilon$, and $u < x$ (statement (vi) implies that $u < x - \varepsilon$) then $x^- \in R^{\varepsilon}_{a,b}(f)$. If $x^+ = b$, $x^+ - x \leqslant \varepsilon$, and $u > x$ (statement (vi) implies that $u > x + \varepsilon$) then $x^+ \in R^{\varepsilon}_{a,b}(f)$. In either case $\tilde{x} \coloneqq x$ satisfies statement (i) and the algorithm terminates.

From statement (iii) and the fact that step 2 set $x \coloneqq (x^+ + x^-)/2$, we see that $x - x^-,\ x^+ - x \leqslant 2^{-p}(b - a)$. If $b - a \leqslant 2\varepsilon$ then the algorithm terminates at step 3 when $p = 1$, otherwise $(b - a > 2\varepsilon)$ $\log_2((b - a)/\varepsilon) > 1$. If $p = \lceil \log_2((b - a)/\varepsilon) \rceil$ then $2^{-p}(b - a) \leqslant \varepsilon$ and the algorithm terminates at step 3. It follows that termination at step 3 implies that $p \leqslant \lceil \max(1, \log_2((b - a)/\varepsilon)) \rceil$.

Step 4: As a result of statement (vi), either $u > x + \varepsilon$ or $u < x - \varepsilon$. Step 4 sets $x^- \coloneqq \min(x^+, (x + u)/2)$ in the first case, or $x^+ \coloneqq \max(x^-, (x + u)/2)$ in the second case. In the first case the following hold:

- $a < x < x^- \leqslant x^+$.
- $f(x^-) \geqslant x^-$ and $R^{\varepsilon}_{a,b}(f) \cap [x^-, b] \neq \emptyset$. This is a result of Theorem 4.2(ii).
- Either $x^+ = b$ or $f(x^+) \leqslant x^+$. This is a result of statement (v).

In the second case the following hold:

- $x^- \leqslant x^+ < x < b$.
- $f(x^+) \leqslant x^+$ and $R^\varepsilon_{a,b}(f) \cap [a, x^+] \neq \emptyset$. This is a result of Theorem 4.2(ii).
- Either $x^- = a$ or $f(x^-) \geqslant x^-$. This is a result of statement (v).

In either case the continuity of f shows that $R^\varepsilon_{a,b}(f) \cap [x^-, x^+] \neq \emptyset$. If $x^- = x^+$ then $\tilde{x} := x^-$ satisfies statement (i) and the algorithm terminates.

We suppose that step 2 will execute again within the current univariate bisection loop, and observe that x^- and x^+ will not change before then. Hence we prove the following statements, which show that statements (ii)–(v) will be true at the beginning of the next execution of step 2.

- $x^- < x^+$. We determined in step 4 that $x^- \leqslant x^+$, and the algorithm terminated in this step if $x^- = x^+$.
- $x^+ - x^- \leqslant 2^{-p}(b-a)$. This follows from statement (ii), the fact that step 2 set $x := (x^+ + x^-)/2$, and the fact that $x < x^- \leqslant x^+$ or $x^- \leqslant x^+ < x$ as a result of the action of step 4.
- $R^\varepsilon_{a,b}(f) \cap [x^-, x^+] \neq \emptyset$. This was proven in the analysis of step 4.
- If $x^- \neq a$ then $f(x^-) \geqslant x^-$. If $x^+ \neq b$ then $f(x^+) \leqslant x^+$. Initially $x^- = a$ and $x^+ = b$. In step 4, x^- is modified only if $u > x + \varepsilon$, in which case $f(x^-) \geqslant x^-$; similarly, x^+ is modified only if $u < x - \varepsilon$, in which case $f(x^+) \leqslant x^+$.

Step 5: Step 5 is not essential to the success of the univariate bisection, but can speed it up. By Theorem 4.2(iii), if $x^- = a$, $x^+ - x^- \leqslant \varepsilon/2$, and $u < x - \varepsilon$ (so that $f(x^+) \leqslant x^+$ by the analysis of step 4) then $a \in R^\varepsilon_{a,b}(f)$. Similarly, if $x^+ = b$, $x^+ - x^- \leqslant \varepsilon/2$, and $u > x + \varepsilon$ (so that $f(x^-) \geqslant x^-$ by the analysis of step 4) then $b \in R^\varepsilon_{a,b}(f)$. In either case the algorithm terminates with a value of $\tilde{x}$ that satisfies statement (i).

Step 6: In step 6, if $x^- \neq a$, $x^+ \neq b$, and $x^+ - x^- \leqslant \varepsilon$ then $f(x^-) \geqslant x^-$ and $f(x^+) \leqslant x^+$ by statement (v), so the continuity of f implies that there exists a fixed point $x^* \in [x^-, x^+]$ of f. Hence by Theorem 4.2(v), $(x^+ + x^-)/2 \in R^\varepsilon_{a,b}(f)$. The algorithm terminates with $\tilde{x} = (x^+ + x^-)/2$ satisfying statement (i).

The analysis of step 4 shows that $x^+ - x^- \leqslant 2^{-p}(b-a)$. If $b - a \leqslant 2\varepsilon$ then the algorithm would have terminated at step 3 when $p = 1$, so $b - a > 2\varepsilon$ and $\log_2((b-a)/\varepsilon) > 1$. If $p = \lceil \log_2((b-a)/\varepsilon) \rceil$ then $2^{-p}(b-a) \leqslant \varepsilon$ and the algorithm terminates at step 6. It follows that termination at step 6 implies that $p \leqslant \lceil \log_2((b-a)/\varepsilon) \rceil$.

Success of the algorithm when $d = 1$: All points of termination of the univariate bisection loop provide a value of $\tilde{x}$ satisfying statement (i). Statement (iii) and the conditions of steps 3 and 6 show that the univariate bisection loop terminates after no more than s function component evaluations.

The case $d > 1$: We next consider the case $d > 1$. As mentioned before, we assume that the hypothesis is true for dimension $d - 1$. We show that it is true for dimension d, i.e., the algorithm returns $\tilde{x}$ satisfying the statement

(vii) $||f(\tilde{x}) - \tilde{x}|| \leqslant \varepsilon$

using no more than $n(d,s)$ function component evaluations.

Step 7: We perform the following analysis when $||b-a|| \leqslant 2\varepsilon$. The action of step 7 in this case is equivalent to setting x to the center of $D_{a,b}$ and returning

$$\tilde{x} := (\max(a_1, \min(b_1, f_1(x))), \ldots, \max(a_d, \min(b_d, f_d(x))));$$

it requires no more than $d \leqslant n(d,s)$ function component evaluations. We note that $\tilde{x} \in D_{a,b}$, hence $||\tilde{x} - x|| \leqslant \varepsilon$. For every $i \in I_d$,

$$|f_i(\tilde{x}) - f_i(x)| \leqslant ||\tilde{x} - x|| \leqslant \varepsilon \tag{8}$$

and

$$|f_i(\tilde{x}) - \tilde{x}_i| = |f_i(\tilde{x}) - \max(a_i, \min(b_i, f_i(x)))|. \tag{9}$$

We consider the following cases in order to show that $|f_i(\tilde{x}) - \tilde{x}_i| \leqslant \varepsilon$, hence $||f(\tilde{x}) - \tilde{x}|| \leqslant \varepsilon$.

- If $f_i(x) \in [a_i, b_i]$ then the right side of (9) becomes $|f_i(\tilde{x}) - f_i(x)|$, so $|f_i(\tilde{x}) - \tilde{x}_i| \leqslant \varepsilon$ by (8).
- If $f_i(x) < a_i$ (the case $f_i(x) > b_i$ is analogous) then $\tilde{x}_i = a_i$ and by (8), $f_i(\tilde{x}) < a_i + \varepsilon$. The range of f is $D^{\varepsilon}_{a,b}$, so $f_i(\tilde{x}) \geqslant a_i - \varepsilon$ and $|f_i(\tilde{x}) - a_i| = |f_i(\tilde{x}) - \tilde{x}_i| \leqslant \varepsilon$.

Step 8: We set $p := 0$ at the beginning of step 8.

The following statements are true at the end of step 8:

(viii) $x \in R^{\varepsilon,S}_{a,b}(f)$, as we now prove. Since g has dimension $d-1$, the recursive call succeeds by our assumption. Since g has domain $D_{\Lambda_S(a),\Lambda_S(b)}$ and f has range $D^{\varepsilon}_{a,b}$, g has range $D^{\varepsilon}_{\Lambda_S(a),\Lambda_S(b)}$. Hence $g \in \mathfrak{F}^{\varepsilon}_{\Lambda_S(a),\Lambda_S(b)}$ and $\Lambda_S(x) = x^* \in R^{\varepsilon}_{\Lambda_S(a),\Lambda_S(b)}(f)$.

(ix) $x_d = (a_d + b_d)/2$.

We will prove in the analysis of steps 12 and 13 that statement (viii) is true at the beginning of each execution of step 9.

The algorithm terminates if $x^-_d = x^+_d$, i.e., if $a_d = b_d$. Under this condition the range of f_d is $[a_d - \varepsilon, a_d + \varepsilon]$, so x is automatically a residual solution of f in the dth component, and $\tilde{x} := x$ satisfies statement (vii).

Multivariate bisection loop

Steps 9–13 constitute the multivariate bisection loop.

Step 9: We set $p := p+1$ at the beginning of step 9.

Step 9 evaluates $u_d := f_d(x)$. If $|u_d - x_d| \leqslant \varepsilon$ then statement (viii) implies that $x \in R^{\varepsilon, S \cup \{d\}}_{a,b}(f)$, so $\tilde{x} := x$ satisfies statement (vii) and the algorithm terminates. Otherwise the following statement holds:

(x) $|u_d - x_d| > \varepsilon$.

We prove that the following statements are true at the beginning of the first execution of step 10 within the current multivariate bisection loop.

(xi) $x_d = (x^+_d + x^-_d)/2$. This is a result of statement (ix).

(xii) $0 < x_d^+ - x_d,\ x_d - x_d^- \leqslant 2^{-p}(b_d - a_d)$. This follows from statement (xi) and the fact that $a_d < b_d$ (step 8).

(xiii) If $x_d^- \neq a_d$ then $||x - x^-|| = x_d - x_d^-$. If $x_d^+ \neq b_d$ then $||x - x^+|| = x_d^+ - x_d$. This holds initially because $x_d^- = a_d$ and $x_d^+ = b_d$.

We will prove in the analysis of steps 12 and 13 that statements (xi)–(xiii) are true at the beginning of each execution of step 10.

Step 10: Step 10 sets $x^- := x$ if $u_d > x_d + \varepsilon$, or $x^+ := x$ if $u_d < x_d - \varepsilon$. This is the only place in the multivariate bisection loop where x^- and x^+ are modified from the values assigned to them in step 8, so the following statements hold:

(xiv) If $x_d^- \neq a_d$ then $f_d(x^-) > x_d^- + \varepsilon$. If $x_d^+ \neq b_d$ then $f_d(x^+) < x_d^+ - \varepsilon$. This follows from statement (x).

(xv) $x_d^- \neq a_d$ or $x_d^+ \neq b_d$. This follows from statement (xii).

We prove that the following statements ((xvi)–(xix)) are true at the end of step 10. Statements (xvi)–(xviii) are consequences of statements (xi)–(xiii) and the fact that step 10 set either x^- or x^+ to x.

(xvi) $x_d^- < x_d^+$.

(xvii) $x_d^+ - x_d^- \leqslant 2^{-p}(b_d - a_d)$.

(xviii) If $x_d^- \neq a_d$ and $x_d^+ \neq b_d$ then $||x^+ - x^-|| = x_d^+ - x_d^-$.

(xix) If $x_d^- \neq a_d$ then $x^- \in R_{a,b}^{\varepsilon,S}(f)$. If $x_d^+ \neq b_d$ then $x^+ \in R_{a,b}^{\varepsilon,S}(f)$. This follows from statement (viii) and the action of step 10.

Step 11: We perform the following analysis of step 11 when $x_d^- = a_d$ (resp. $x_d^+ = b_d$) and $x_d^+ - x_d^- \leqslant \varepsilon$. By statements (xiv), (xv), and (xix), $f_d(x^+) < x_d^+ - \varepsilon$ (resp. $f_d(x^-) > x_d^- + \varepsilon$) and $x^+ \in R_{a,b}^{\varepsilon,S}(f)$ (resp. $x^- \in R_{a,b}^{\varepsilon,S}(f)$). Theorem 4.4 applied to f, S, x^+ (resp. x^-), and $y := a_d$ (resp. $y := b_d$) yields definitions of a', b', and $g \in \mathfrak{F}_{a',b'}^{\varepsilon}$. By Theorem 4.4, $||b' - a'|| \leqslant 2(x_d^+ - x_d^-) \leqslant 2\varepsilon$. The algorithm recursively executes itself on g to obtain $x^* \in R_{a',b'}^{\varepsilon}(g)$. (Since g has dimension $d-1$, the recursive call succeeds by our assumption.) We set $x := \Psi_S(x^*, y)$. By Theorem 4.4, $x \in R_{a,b}^{\varepsilon,S}(f)$ and $||\Lambda_S(x - x^+)|| \leqslant |x_d - x_d^+|$ (resp. $||\Lambda_S(x - x^-)|| \leqslant |x_d - x_d^-|$), hence

$$||x - x^+|| = |x_d - x_d^+| \quad (\text{resp. } ||x - x^-|| = |x_d - x_d^+|). \tag{10}$$

We define $z^- \equiv x$ and $z^+ \equiv x^+$ (resp. $z^- \equiv x^-$ and $z^+ \equiv x$). We define the function $f' : [a_d, b_d] \to [a_d - \varepsilon, b_d + \varepsilon]$ as

$$f'(c) \equiv \begin{cases} f_d(z^-[d \leftarrow c]), & c \in [a_d, z_d^-], \\ f_d\left(z^- + \frac{c - z_d^-}{z_d^+ - z_d^-}(z^+ - z^-)\right), & c \in [z_d^-, z_d^+], \\ f_d(z^+[d \leftarrow c]), & c \in [z_d^+, b_d]. \end{cases}$$

We determine that $f' \in \mathfrak{F}_{a_d,b_d}^{\varepsilon}$ by applying Lemma 4.3 with L equal to the union of the line segments $[z^-[d \leftarrow a_d], z^-]$, $[z^-, z^+]$, and $[z^+, z^+[d \leftarrow b_d]]$ (L satisfies the criteria of Lemma 4.3 by (10)). Theorem 4.2(iv) applied to f' shows that $x \in R_{a,b}^{\varepsilon}(f)$, so $\tilde{x} := x$ satisfies statement (vii).

Statement (xvii) shows that $x_d^+ - x_d^- \leqslant 2^{-p}(b_d - a_d)$. If $b_d - a_d \leqslant 2\varepsilon$ then the algorithm terminates at step 11 when $p = 1$, otherwise $(b_d - a_d > 2\varepsilon)$ $\log_2((b_d - a_d)/\varepsilon) > 1$. If $p = \lceil \log_2((b_d - a_d)/\varepsilon) \rceil$ then $2^{-p}(b_d - a_d) \leqslant \varepsilon$ and the algorithm terminates at step 11. It follows that termination at step 11 implies that $p \leqslant \lceil \max(1, \log_2((b_d - a_d)/\varepsilon)) \rceil$.

Step 12: We perform the following analysis of step 12 when $x_d^- = a_d$ (resp. $x_d^+ = b_d$). As a result of statements (xv) and (xix), $x^+ \in R_{a,b}^{\varepsilon,S}(f)$ (resp. $x^- \in R_{a,b}^{\varepsilon,S}(f)$). Theorem 4.4 applied to f, S, x^+ (resp. x^-), and $y := (x_d^+ + x_d^-)/2$ yields definitions of a', b', and $g \in \mathfrak{F}_{a',b'}^{\varepsilon}$. By Theorem 4.4 and statement (xvii),

$$||b' - a'|| \leqslant 2 \cdot \tfrac{1}{2}(x_d^+ - x_d^-) \leqslant 2^{-p}(b_d - a_d).$$

The algorithm recursively executes itself on g to obtain $x^* \in R_{a',b'}^{\varepsilon}(g)$. (Since g has dimension $d - 1$, the recursive call succeeds by our assumption.) We set $x := \Psi_S(x^*, y)$. By Theorem 4.4, $x \in R_{a,b}^{\varepsilon,S}(f)$ and $||\Lambda_S(x - x^+)|| \leqslant |x_d - x_d^+|$ (resp. $||\Lambda_S(x - x^-)|| \leqslant |x_d - x_d^-|$), so that $||x - x^+|| = x_d^+ - x_d$ (resp. $||x - x^-|| = x_d - x_d^-$).

We have shown that $x \in R_{a,b}^{\varepsilon,S}(f)$ at this point, so that if step 9 is executed again within the current multivariate bisection loop, statement (viii) will be true at the beginning of its next execution.

We suppose that step 10 will be executed again within the current multivariate bisection loop. The analysis of step 12 proves the following statements, which show that statements (xi)–(xiii) will be true at the beginning of the next execution of step 10.

- $x_d = (x_d^+ + x_d^-)/2$.
- $0 < x_d^+ - x_d,\ x_d - x_d^- \leqslant 2^{-(p+1)}(b_d - a_d)$. This follows from statements (xvi) and (xvii), and the fact that $x_d = (x_d^+ + x_d^-)/2$. This implies that $x_d \neq a_d$ and $x_d \neq b_d$.
- If $x_d^- \neq a_d$ then $||x - x^-|| = x_d - x_d^-$. If $x_d^+ \neq b_d$ then $||x - x^+|| = x_d^+ - x_d$.

Step 13: If we have arrived at step 13 then we know that $x_d^- \neq a_d$ and $x_d^+ \neq b_d$. By statement (xix), x^+, $x^- \in R_{a,b}^{\varepsilon,S}(f)$. We apply Theorem 4.4 to f, S, x^-, and $y := (x_d^+ + x_d^-)/2$ to obtain definitions $a^- \equiv a'$, $b^- \equiv b'$, and $g^- \equiv g \in \mathfrak{F}_{a',b'}^{\varepsilon}$. We apply Theorem 4.4 to f, S, x^+, and y to obtain definitions $a^+ \equiv a'$, $b^+ \equiv b'$, and $g^+ \equiv g \in \mathfrak{F}_{a',b'}^{\varepsilon}$. By Theorem 4.4 and statement (xvii), $||b^+ - a^+||$, $||b^- - a^-|| \leqslant 2 \cdot \frac{1}{2}(x_d^+ - x_d^-) \leqslant 2^{-p}(b_d - a_d)$. We define $a^*, b^* \in \mathbb{R}^{d-1}$ as $a_i^* = \max(a_i^+, a_i^-)$ and $b_i^* = \min(b_i^+, b_i^-)$ for $i = 1, \ldots, d-1$. We use the fact that $||x^+ - x^-|| = x_d^+ - x_d^-$ (statement (xviii)) to show that $a_i^* \leqslant b_i^*$ for $i = 1, \ldots, d-1$; as a result, $D_{a^*,b^*} = D_{a^+,b^+} \cap D_{a^-,b^-} \neq \emptyset$ and $||b^* - a^*|| \leqslant 2^{-p}(b_d - a_d)$. This reduces to showing that $a_i^+ \leqslant b_i^-$ (analogously, $a_i^- \leqslant b_i^+$) for $i = 1, \ldots, d-1$. For arbitrary $i \in S$, if $x_i^+ \leqslant x_i^-$ then $a_i^+ \leqslant x_i^+ \leqslant x_i^- \leqslant b_i^-$, so we assume that $x_i^+ > x_i^-$; this yields $0 < x_i^+ - x_i^- \leqslant x_d^+ - x_d^-$

(statement (xviii)) and $x_i^+ - \frac{1}{2}(x_d^+ - x_d^-) \leqslant x_i^- + \frac{1}{2}(x_d^+ - x_d^-)$. It follows that

$$a_i^+ = \max\left(a_i, x_i^+ - \tfrac{1}{2}(x_d^+ - x_d^-)\right) \leqslant \min\left(b_i, x_i^- + \tfrac{1}{2}(x_d^+ - x_d^-)\right) = b_i^-.$$

We observe that for all $z \in D_{a^*,b^*}$,

$$g^+(z) = g^-(z) \in D^{\varepsilon}_{a^+,b^+} \cap D^{\varepsilon}_{a^-,b^-} = D^{\varepsilon}_{a^*,b^*},$$

hence the function $g \equiv g^+ = g^-$ having domain D_{a^*,b^*} is in $\mathfrak{F}^{\varepsilon}_{a^*,b^*}$. The algorithm recursively executes itself on g to obtain $x^* \in R^{\varepsilon}_{a^*,b^*}(g)$. (Since g has dimension $d-1$, the recursive call succeeds by our assumption.) We set $x := \Psi_S(x^*, y)$. By Theorem 4.4, $x \in R^{\varepsilon,S}_{a,b}(f)$, $\|\Lambda_S(x - x^+)\| \leqslant |x_d - x_d^+|$, and $\|\Lambda_S(x - x^-)\| \leqslant |x_d - x_d^-|$, so that

$$\|x - x^-\| = x_d - x_d^- \quad \text{and} \quad \|x - x^+\| = x_d^+ - x_d. \tag{11}$$

We now show that if $x_d^+ - x_d^- \leqslant 2\varepsilon$ then $x \in R^{\varepsilon}_{a,b}(f)$, hence $\tilde{x} := x$ satisfies statement (vii). We define the function f' on $[a_d, b_d]$ as

$$f'(c) \equiv \begin{cases} f_d(x^-[d \leftarrow c]), & c \in [a_d, x_d^-], \\ f_d\left(x^- + \frac{c - x_d^-}{x_d - x_d^-}(x - x^-)\right), & c \in [x_d^-, x_d], \\ f_d\left(x + \frac{c - x_d}{x_d^+ - x_d}(x^+ - x)\right), & c \in [x_d, x_d^+], \\ f_d(x^+[d \leftarrow c]), & c \in [x_d^+, b_d]. \end{cases}$$

We determine that $f' \in \mathfrak{F}^{\varepsilon}_{a_d,b_d}$ by applying Lemma 4.3 with L equal to the union of the line segments $[x^-[d \leftarrow a_d], x^-]$, $[x^-, x^+]$, and $[x^+, x^+[d \leftarrow b_d]]$ (L satisfies the criteria of Lemma 4.3 by (11)). Statement (xiv) implies that $f'(x_d^-) > x_d^- + \varepsilon$ and $f'(x_d^+) < x_d^+ - \varepsilon$, so by continuity, f' has a fixed point c^* in the interval $[x_d^-, x_d^+]$. Furthermore, statement (xiv) combined with the Lipschitz continuity of f' implies that $c^* \notin [x_d^-, x_d^- + \varepsilon/2] \cup [x_d^+ - \varepsilon/2, x_d^+]$. It follows that $c^* \in (x_d^- + \varepsilon/2, x_d^+ - \varepsilon/2)$ and $|c^* - x_d| < \varepsilon/2$. Theorem 4.2(v) now yields $|f'(x_d) - x_d| = |f_d(x) - x_d| \leqslant \varepsilon$.

Statement (xvii) shows that $x_d^+ - x_d^- \leqslant 2^{-p}(b_d - a_d)$. If $b_d - a_d \leqslant 2\varepsilon$ then the algorithm would have terminated at step 11 when $p = 1$, so $b_d - a_d > 2\varepsilon$ and $\log_2((b_d - a_d)/\varepsilon) > 1$. If $p = \lceil \log_2((b_d - a_d)/\varepsilon) \rceil - 1$ then $2^{-p}(b_d - a_d) \leqslant 2\varepsilon$ and the algorithm terminates at step 13. It follows that termination at step 13 implies that $p \leqslant \lceil \log_2((b_d - a_d)/\varepsilon) \rceil - 1$.

For the remainder of the analysis of step 13 we assume that $x_d^+ - x_d^- > 2\varepsilon$. We have shown that $x \in R^{\varepsilon,S}_{a,b}(f)$ at this point, so that if step 9 is executed again within the current multivariate bisection loop, statement (viii) will be true at the beginning of its next execution.

We suppose that step 10 will be executed again within the current multivariate bisection loop. The analysis of step 13 proves the following statements, which show that statements (xi)–(xiii) will be true at the beginning of the next execution of step 10.

- $x_d = (x_d^+ + x_d^-)/2$.

- $0 < x_d^+ - x_d,\ x_d - x_d^- \leqslant 2^{-(p+1)}(b_d - a_d)$. This follows from statements (xvi) and (xvii), and the fact that $x_d = (x_d^+ + x_d^-)/2$. This implies that $x_d \neq a_d$ and $x_d \neq b_d$.
- $||x - x^-|| = x_d - x_d^-$ and $||x - x^+|| = x_d^+ - x_d$.

Success of the algorithm when $d > 1$: All points of termination of the multivariate bisection loop provide a value of $\tilde{x}$ satisfying statement (vii). Statement (xvii) and the conditions of steps 7, 11, and 13 show that the multivariate bisection loop terminates after no more than $\lceil \log_2((b_d - a_d)/\varepsilon) \rceil \leqslant s$ evaluations of f_d. The number of recursive calls executed during the loop does not exceed the number of evaluations of f_d by more than one, so this number is bounded above by s. By our assumption, the number of function component evaluations performed by each recursive call is bounded by $\sum_{i=1}^{d-1} \theta^i$, where $\theta \equiv \lceil \max(1, \log_2(||\Lambda_S(b-a)||/\varepsilon)) \rceil + 1 \leqslant s$ and $S \equiv \{1, \dots, d-1\}$. Hence the total number of function component evaluations executed by the multivariate bisection loop is bounded above by

$$s \sum_{i=1}^{d-1} \theta^i + s \leqslant s \sum_{i=1}^{d-1} s^i + s = n(d, s).$$

This concludes the proof of Theorem 5.1. $\square$

5.3. *Worst-case complexity of PFix*

In this section we describe the worst-case complexity of the PFix algorithm for the residual criterion when the Lipschitz constant $q = 1$, and for the absolute criterion when $q < 1$. We also show that when $q > 1$, the absolute criterion problem has infinite worst-case complexity when information consists of function evaluations.

5.3.1. *Residual criterion,* $q = 1$

Theorem 5.1 proved that for a general domain $[a, b] \in \mathbb{R}^d$, $d \geqslant 1$, and a function $f \in \mathfrak{F}_{a,b}^{\varepsilon}$, PFix computes a point in $R_{a,b}^{\varepsilon}(f)$ using no more than $n(d, s)$ function component evaluations, where n and s are defined in (3).

For the case $[a, b] = [0, 1]^d$ and $0 < \varepsilon < 0.5$, Theorem B.1 provides a tighter worst-case complexity bound. In this case PFix computes a residual solution using no more than $B(d, r)$ function component evaluations, where B and r are defined in (4).

5.3.2. *Absolute criterion,* $q < 1$

For a function $f \in \mathfrak{F}^d$ satisfying (6) with $0 < q < 1$, we show how to compute a fixed point approximation $\tilde{x} \in D^d$ satisfying (7). The Banach theorem shows that f has a unique fixed point x^*. For every $x \in D^d$,

$$||x - x^*|| \leqslant ||x - f(x)|| + ||f(x) - f(x^*)|| \leqslant ||x - f(x)|| + q||x - x^*||$$

so that

$$||x - x^*|| \leqslant \frac{1}{1-q} ||x - f(x)||.$$

Hence executing the algorithm with tolerance $(1-q)\varepsilon$ yields the desired $\tilde{x}$.

5.3.4. Absolute criterion, $q > 1$

We show that the problem of satisfying the absolute criterion (7) when $q > 1$ has infinite worst-case complexity in the class of algorithms whose information consists of arbitrary sequential function evaluations. Our proof is similar to the proof given for the 2-norm case in [18], which also handles $q = 1$. We assume that $0 < \varepsilon < 0.5$. It suffices to consider the case $d = 2$. Suppose we have an adaptive sequence of $n > 0$ evaluation points in $[0,1]^2$,

$$\{x^1, x^2(x^1, f^1), x^3(x^1, f^1; x^2, f^2), \ldots, x^n(x^1, f^1; \ldots; x^{n-1}, f^{n-1})\},$$

where $f^1, \ldots, f^n$ are the function values that will be assigned at points $x^1, \ldots, x^n$. We define the class $\mathfrak{F}^{q,2}$ of functions on $[0,1]^2$ that are Lipschitz continuous with constant q. We will show that there exist two functions $f^+, f^- \in \mathfrak{F}^{q,2}$ such that $f^+(x^i) = f^-(x^i) = f^i$ for $i = 1, \ldots, n$, f^+ has a unique fixed point x^+, f^- has a unique fixed point x^-, and $||x^+ - x^-|| = 1 > 2\varepsilon$. Hence there exists no solution satisfying the absolute criterion for both f^+ and f^-. Infinite worst-case complexity follows from the fact that $n > 0$ is arbitrary. It suffices to consider the case $1 < q < q'$ for some $q' > 1$; we take $q' = 2$.

We choose a positive constant $\delta < 2^{-(n+1)}(q-1)$, set $l^0 = 0$ and $r^0 = 1$, and assign f^i, $i = 1, \ldots, n$, using the following steps.

1. Set $i := 1$. Generate an arbitrary $x^1 \in [0,1]^2$.
2. If $x_1^i \leqslant l^{i-1}$ then assign $f^i := (x_1^i + \delta, \delta + (1-2\delta)x_2^i)$ and go to step 6.
3. If $x_1^i \geqslant r^{i-1}$ then assign $f^i := (x_1^i - \delta, \delta + (1-2\delta)x_2^i)$ and go to step 6.
4. Set $m := (l^{i-1} + r^{i-1})/2$. If $x_1^i \leqslant m$ then assign $f^i := (x_1^i + \delta, \delta + (1-2\delta)x_2^i)$, set $l^i := x_1^i$, set $r^i := r^{i-1}$, and go to step 7.
5. Assign $f^i := (x_1^i - \delta, \delta + (1-2\delta)x_2^i)$, set $l^i := l^{i-1}$, set $r^i := x_1^i$, and go to step 7.
6. Set $l^i := l^{i-1}$ and $r^i := r^{i-1}$.
7. If $i := n$ then terminate, otherwise set $i := i + 1$, generate an arbitrary $x^i \in [0,1]^2$, and go to step 2.

At this point we know that $r^n - l^n \geqslant 2^{-n}$ and that for $i = 1, \ldots, n$, $x_1^i \notin (l^n, r^n)$ and

$$f^i = \begin{cases} (x_1^i + \delta, \delta + (1-2\delta)x_2^i), & x_1^i \leqslant l^n, \\ (x_1^i - \delta, \delta + (1-2\delta)x_2^i), & x_1^i \geqslant r^n. \end{cases}$$

We define $l \equiv l^n$, $r \equiv r^n$, and $m \equiv (l+r)/2$; clearly $\delta < 2^{-(n+1)} < m-l = r-m$. We define the function $f^+ \in \mathfrak{F}^{q,2}$ as

$$f_1^+(x) \equiv \begin{cases} x_1 + \delta, & x_1 \leqslant l, \\ x_1 + \frac{x_1 - m}{l-m}\delta, & l \leqslant x_1 \leqslant r, \\ x_1 - \delta, & x_1 \geqslant r \end{cases}$$

and

$$f_2^+(x) \equiv \begin{cases} \delta + (1-2\delta)x_2, & x_1 \leqslant l + x_2(m-l), \\ \delta + (1-\delta)x_2 - \frac{x_1 - l}{m-l}\delta, & l + x_2(m-l) \leqslant x_1 \leqslant m, \\ \delta + (1-\delta)x_2 - \frac{x_1 - r}{m-r}\delta, & m \leqslant x_1 \leqslant r - x_2(r-m), \\ \delta + (1-2\delta)x_2, & x_1 \geqslant r - x_2(r-m), \end{cases}$$

ad the function $f^- \in \mathfrak{F}^{q,2}$ as

$$f^-(x) \equiv (f_1^+(x), 1 - f_2^+((x_1, 1 - x_2))).$$

Figs. 4 and 5 illustrate the functions f_1^+ and f_2^+, respectively. Clearly $f^+(x^i) = f^-(x^i) = f^i$ for $i = 1, \ldots, n, f^+$ has a unique fixed point at $(m, 0)$, and f^- has a unique fixed point at $(m, 1)$. We easily see that f^+ and f^- map $[0,1]^2$ into $[0,1]^2$.

We show that f^+ is Lipschitz continuous with constant q. We observe that if we divide a domain into closed convex subdomains with nonintersecting interiors, and a function is Lipschitz continuous with constant q on each subdomain, then it is Lipschitz continuous with constant q on the entire domain. Suppose we have $u, v \in [0,1]^2$. We consider f_1^+:

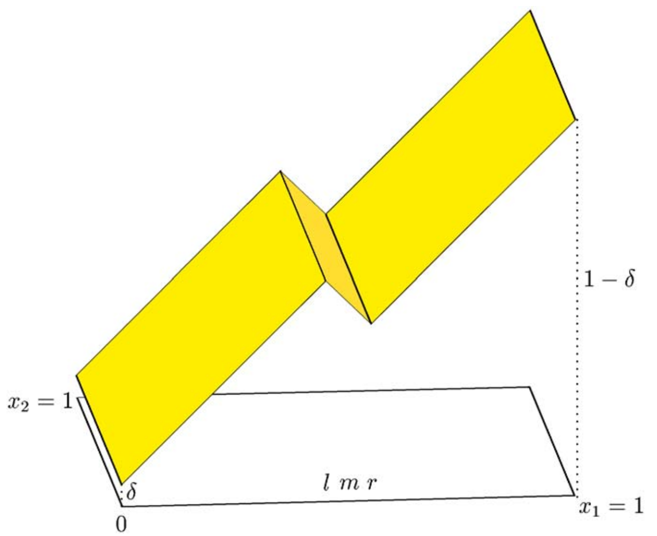

Fig. 4. The function f_1^+.

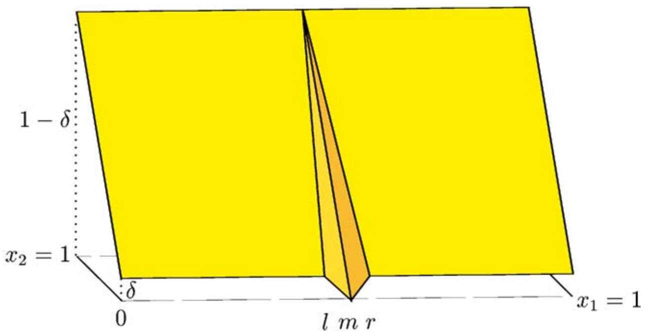

Fig. 5. The function f_2^+.

- If $u, v \in \{x\colon\ x_1 \leqslant l\}$ (the case $x_1 \geqslant r$ is analogous) then $|f_1^+(u) - f_1^+(v)| = |u_1 + \delta - v_1 - \delta| = |u_1 - v_1|$.
- If $u, v \in \{x\colon\ l \leqslant x_1 \leqslant r\}$ then since $\delta < m - l$,

$$|f_1^+(u) - f_1^+(v)| = \left|\left(1 + \frac{1}{l-m}\delta\right)u_1 - \left(1 + \frac{1}{l-m}\delta\right)v_1\right| < |u_1 - v_1|.$$

Hence f_1^+ is Lipschitz continuous with constant 1. We consider f_2^+:

- If $u, v \in \{x\colon\ x_1 \leqslant l + x_2(m-l)\}$ (the case $x_1 \geqslant r - x_2(r-m)$ is analogous) then since $\delta < 1/2$,

$$\begin{aligned}|f_2^+(u) - f_2^+(v)| &= |\delta + (1-2\delta)u_2 - \delta - (1-2\delta)v_2| \\ &= (1-2\delta)|u_2 - v_2| < |u_2 - v_2|.\end{aligned}$$

- If $u, v \in \{x\colon\ l + x_2(m-l) \leqslant x_1 \leqslant m\}$ (the case $m \leqslant x_1 \leqslant r - x_2(r-m)$ is analogous) then since $\delta < 2^{-(n+1)}(q-1)$ and $m - l > 2^{-(n+1)}$,

$$\begin{aligned}|f_2^+(u) - f_2^+(v)| &= \left|(1-\delta)(u_2 - v_2) - \frac{\delta}{m-l}(u_1 - v_1)\right| \\ &\leqslant \left(1 - \delta + \frac{\delta}{m-l}\right)||u - v|| < q||u - v||.\end{aligned}$$

Finally we show that f^- is Lipschitz continuous with constant q. For arbitrary $u, v \in [0,1]^2$, since $f_1^+(x, y) = f_1^+(x, 1-y)$ for all $x, y \in [0,1]^2$,

$$\begin{aligned}||f^-(u) - f^-(v)|| &= \max(|f_1^+(u) - f_1^+(v)|, |f_2^+((u_1, 1-u_2)) - f_2^+((v_1, 1-v_2))|) \\ &= \max(|f_1^+((u_1, 1-u_2)) - f_1^+((v_1, 1-v_2))|, \\ &\qquad |f_2^+((u_1, 1-u_2)) - f_2^+((v_1, 1-v_2))|) \\ &= ||f^+((u_1, 1-u_2)) - f^+((v_1, 1-v_2))|| \\ &\leqslant q\max(|u_1 - v_1|, |u_2 - v_2|) = q||u - v||.\end{aligned}$$

	Number of evaluations	Upper bound	Ratio
$d = 6$, $\epsilon = 10^{-13}$	938,168	16,085,280	0.058
$d = 35$, $\epsilon = 0.00001$	276,701,923	31,792,967,748,570	8.7×10^{-6}
$d = 100$, $\epsilon = 0.001$	622,979,026	85,620,455,854,791	7.3×10^{-6}
$d = 1000$, $\epsilon = 0.025$	168,917,001	2,819,723,266,110,751	6×10^{-8}

Fig. 6. Test results for $f^1(x)$.

	Number of evaluations	Upper bound	Ratio
$d = 6$, $\epsilon = 10^{-13}$	1,502	16,085,280	9.3×10^{-5}
$d = 35$, $\epsilon = 0.00001$	10,226,122	31,792,967,748,570	3.2×10^{-7}
$d = 100$, $\epsilon = 0.001$	79,964,874	85,620,455,854,791	9.3×10^{-7}
$d = 1000$, $\epsilon = 0.025$	1,351,316,012	2,819,723,266,110,751	4.8×10^{-7}

Fig. 7. Test results for $f^2(x)$.

	Number of evaluations	Upper bound	Ratio
$d = 6$, $\epsilon = 10^{-13}$	6,022,868	16,085,280	0.37
$d = 35$, $\epsilon = 0.00001$	12,220,729,709	31,792,967,748,570	3.8×10^{-4}
$d = 100$, $\epsilon = 0.001$	3,709,516,968	85,620,455,854,791	4.3×10^{-5}
$d = 1000$, $\epsilon = 0.025$	42,752,957,249	2,819,723,266,110,751	1.5×10^{-5}

Fig. 8. Test results for $f^3(x)$.

6. Numerical results

In Figs. 6–8 we emphasize that "number of evaluations" refers to evaluations of function components, i.e., $f_1, f_2, \ldots$ are counted separately. We use $B(d, \lceil \log_2(1/\varepsilon) \rceil)$ as an upper complexity bound (see Theorem B.1).

Fig. 6 summarizes the test results for the function $f^1 : [0,1]^d \to [0,1]^d$ defined for $i = 1, \ldots, d$ as

$$f_i^1(x) \equiv \begin{cases} 0.1 + \frac{1}{3}\ln((x_i+1)(x_{(i \bmod (d-1))+1}+1)(x_{((i+1) \bmod (d-1))+1}+1)), & i \text{ even}, \\ 0.4 + \frac{1}{3}\sin(x_i + x_{(i \bmod (d-1))+1} + x_{((i+1) \bmod (d-1))+1}), & i \text{ odd}. \end{cases}$$

We show in Appendix C that f^1 satisfies the necessary Lipschitz criterion when $d > 0$. It is clear that the range of f^1 is a subset of $[0,1]^d$.

Fig. 7 summarizes the test results for the function $f^2 : [0,1]^d \to [0,1]^d$ defined for $i = 1, \ldots, d$ as

$$f_i^2(x) \equiv \max(0, 1 - ||x - y^i||),$$

$$y_j^i \equiv \frac{1}{2} - \frac{(2(i-1)-d)(2(j-1)-d)}{2d^2}, \quad j = 1, \ldots, d.$$

These results show that the actual number of evaluations performed is often many orders of magnitude less than the complexity bound.

Fig. 8 summarizes the test results for the function $f^3 : [0,1]^d \to [0,1]^d$ defined as

$$f^3(x) \equiv 0,$$

$d=5$	$\epsilon=10^{-3}$ $q=0.99$	$\epsilon=10^{-5}$ $q=0.9999$	$\epsilon=10^{-7}$ $q=0.999999$
PFix	376	737	1,088
SI	3,440	575,260	80,590,440

$d=10$	$\epsilon=10^{-3}$ $q=0.99$	$\epsilon=10^{-5}$ $q=0.9999$	$\epsilon=10^{-7}$ $q=0.999999$
PFix	6,889	17,880	31,524
SI	6,880	1,151,240	161,180,880

$d=25$	$\epsilon=10^{-3}$ $q=0.99$	$\epsilon=10^{-5}$ $q=0.9999$	$\epsilon=10^{-7}$ $q=0.999999$
PFix	553,388	4,438,344	49,973,032
SI	17,200	2,878,100	402,952,200

Fig. 9. Test results for $f^4(x)$, absolute criterion.

they also apply to any function that happens to return zero for all evaluations performed by the algorithm.

Fig. 9 compares the performance of the PFix algorithm with that of simple iteration (SI) on the function $f^4 : [0,1]^d \to [0,1]^d$, defined for $i = 1, \ldots, d$ as

$$f_i^4(x) \equiv \max(0, 1 - q||x - y^i||),$$

$$y_j^i \equiv \frac{1}{2} - \frac{(2(i-1)-d)(2(j-1)-d)}{2d^2}, \quad j = 1, \ldots, d.$$

Here q is a Lipschitz constant $(0<q<1)$ and the PFix and SI methods compute solutions satisfying the absolute error criterion. The PFix method does so by using the error tolerance $\varepsilon(1-q)$ as described in Section 5.3.2, while the SI method calls $x^{k+1} = f(x^k)$ (with $x^0 = (0.5, \ldots, 0.5)$) up to $\lceil \log \varepsilon / \log q \rceil$ times, stopping if $||x^k - f(x^k)|| \leqslant \varepsilon(1-q)$. The numbers shown indicate evaluations of function components in both cases. The results indicate that PFix frequently outperforms SI for small dimensions and q close to 1.

7. Future work

We expect to prove that the worst-case complexity of computing an ε-absolute approximation $\tilde{x} : ||\tilde{x} - \alpha||_\infty \leqslant \varepsilon$ for $q = 1$ is infinite.

We believe that it is possible to improve PFix by replacing univariate bisection with the BEDFix algorithm [16] in the univariate bisection loop, and with the BEFix algorithm [17] in the multivariate bisection loop. These bivariate algorithms outperform PFix when $d = 2$. We expect that this approach will achieve a complexity bound of $\mathcal{O}(r^{\lceil d/2 \rceil}/\lceil d/2 \rceil!)$ as $r \to \infty$ and $\mathcal{O}(\lceil d/2 \rceil^{r+1}/(r+1)!)$ as $d \to \infty$ (here $r \equiv \lceil \log_2(1/\varepsilon) \rceil$).

We continue to pursue a method for the residual case that has a complexity bound of the form $c(d)\log_2(1/\varepsilon)$, where $c(d)$ is a polynomial of small degree in the dimension d. We expect to prove that the optimal worst-case complexity of the problem is $d\log_2(1/\varepsilon)$.

We are investigating the possibility of a quantum computer algorithm that will operate on d qubits and have worst-case complexity independent of the dimension (i.e., $O(\log_2(1/\varepsilon))$).

Acknowledgments

We thank the referees for comments and suggestions that significantly improved our presentation.

Appendix A. Summation identities

In this appendix we prove several identities that simplify repeated summation expressions. For integers d, l, m we define the function

$$\sigma(d,l,m) \equiv \begin{cases} 0, & l>m \text{ or } d<0, \\ 1, & l\leqslant m \text{ and } d=0, \\ \sum\limits_{i=l}^{m} \sigma(d-1,l,i), & l\leqslant m \text{ and } d>0 \end{cases}$$

and for integers $m\geqslant 1$ and d the function

$$C(d,m) \equiv \begin{cases} 0, & d<0, \\ \binom{d+m-1}{m-1}, & d\geqslant 0. \end{cases}$$

Lemma A.1. *For all integers l, $m\geqslant l$, and d, $\sigma(d,l,m+1) = \sum_{i=0}^{d}\sigma(i,l,m)$.*

Proof. Clearly the hypothesis is true when $d\leqslant 0$. When $d=1$,

$$\sigma(1,l,m+1) = \sum_{i=l}^{m+1} 1 = m+2-l = \sum_{i=l}^{m} 1 + 1 = \sum_{i=0}^{1} \sigma(i,l,m).$$

Given $d\geqslant 1$ we assume inductively that the hypothesis is true for d. It follows that

$$\begin{aligned}\sigma(d+1,l,m+1) &= \sum_{i=l}^{m+1} \sigma(d,l,i) = \sum_{i=l}^{m} \sigma(d,l,i) + \sigma(d,l,m+1) \\ &= \sigma(d+1,l,m) + \sum_{i=0}^{d} \sigma(i,l,m) = \sum_{i=0}^{d+1} \sigma(i,l,m). \qquad \square\end{aligned}$$

Lemma A.2. *For all integers $m\geqslant 1$ and d, $C(d,m+1) = \sum_{i=0}^{d} C(i,m)$.*

Proof. Clearly the hypothesis is true when $d \leqslant 0$. When $d = 1$,

$$C(1, m+1) = \binom{m+1}{m} = m + 1 = \binom{m}{m-1} + 1 = \sum_{i=0}^{1} C(i, m).$$

Given $d \geqslant 1$ we assume inductively that the hypothesis is true for d. It follows that

$$\begin{aligned} C(d+1, m+1) &= \binom{d+m+1}{m} = \binom{d+m}{m}\left(1 + \frac{m}{d+1}\right) \\ &= C(d, m+1) + \binom{d+m}{m-1} = \sum_{i=0}^{d+1} C(i, m). \qquad \square \end{aligned}$$

Lemma A.3. *For all integers d and for* $m \geqslant 1$, $C(d, m) = \sigma(d, 1, m)$.

Proof. Clearly the hypothesis is true when $d < 0$, and $d = 0$ implies that

$$\binom{d+m-1}{m-1} = \frac{(m-1)!}{0!(m-1)!} = 1 = \sigma(0, 1, m).$$

Setting $m = 1$ we see that for all $d > 0$,

$$C(d, 1) = \frac{d!}{0!d!} = 1 = \sigma(0, 1, 1) = \sum_{i=1}^{1} \sigma(0, 1, i) = \sigma(1, 1, 1) = \cdots = \sigma(d, 1, 1).$$

We assume inductively that for some $m \geqslant 1$, $C(i, m) = \sigma(i, 1, m)$ for all integers $i > 0$. We choose an arbitrary $d > 0$ and use Lemmas A.1 and A.2 to obtain

$$\sigma(d, 1, m+1) = \sum_{i=0}^{d} \sigma(i, 1, m) = \sum_{i=0}^{d} C(i, m) = C(d, m+1). \qquad \square \tag{12}$$

Lemma A.4. *For all integers d and for* $m \geqslant 2$, $\sigma(d, 2, m) = \sigma(d, 1, m) - \sigma(d-1, 1, m)$.

Proof. Clearly the hypothesis is true when $d \leqslant 0$. When $d = 1$,

$$\sigma(1, 2, m) = \sum_{i=2}^{m} 1 = m - 1 = \sum_{i=1}^{m} 1 - 1 = \sigma(1, 1, m) - \sigma(0, 1, m).$$

Given $d \geqslant 1$ we assume inductively that the hypothesis is true for d, i.e., $\sigma(d, 2, i) = \sigma(d, 1, i) - \sigma(d-1, 1, i)$ for all $i \geqslant 2$. For arbitrary $m \geqslant 2$ we obtain

$$\begin{aligned} \sigma(d+1, 2, m) &= \sum_{i=2}^{m} \sigma(d, 2, i) = \sum_{i=2}^{m} (\sigma(d, 1, i) - \sigma(d-1, 1, i)) \\ &= \sigma(d+1, 1, m) - \sigma(d, 1, m) - \sigma(d, 1, 1) + \sigma(d-1, 1, 1) \\ &= \sigma(d+1, 1, m) - \sigma(d, 1, m). \qquad \square \end{aligned}$$

Appendix B. Complexity result

Given $d \geqslant 1$ and $a, b \in \mathbb{R}^d$, we define $N(d, a, b)$ as the supremum over all $f \in \mathfrak{F}^{\varepsilon}_{a,b}$ of the number of function component evaluations that PFix performs when computing a residual solution for f. In the following theorem we determine an upper bound on N when $D_{a,b} = [0, 1]^d$; the complexity bound (4) is a result. We make use of the functions σ and C defined in Appendix A.

Theorem B.1. *For* $d \geqslant 1$ *and* $0 < \varepsilon < 0.5$, $N(d, 0, (1, \ldots, 1)) \leqslant B(d, r) \equiv C(d, r) - C(d - 1, r) + 2(C(d - 1, r + 2) - C(d - 2, r + 2))$, *where* $r \equiv \lceil \log_2(1/\varepsilon) \rceil$.

Proof. We proceed by defining a function U that depends on the permissible domains $[a, b]$, showing that $N \leqslant U$ for all such domains, and proving the hypothesis for U in place of N.

For all $c^1, c^2 \in \mathbb{R}$ such that $c^1 \leqslant c^2$ we define

$$r(c^1, c^2) \equiv \lceil \max(1, \log_2((c^2 - c^1)/\varepsilon)) \rceil$$

and observe that $r \equiv r(0, 1)$ since $\varepsilon < 0.5$. Given $d \geqslant 1$ and $a, b \in \mathbb{R}^d$ such that $a_i \leqslant b_i$ for $i = 1, \ldots, d$, we define

$$S \equiv \begin{cases} \emptyset, & d = 1, \\ \{1, \ldots, d - 1\}, & d > 1 \end{cases}$$

and the function

$$U(d, a, b) \equiv \begin{cases} r(a_d, b_d), & d = 1, \\ d, & d > 1 \text{ and } \|b - a\| \leqslant 2\varepsilon, \\ U(d - 1, 0, \Lambda_S(b - a)) + \displaystyle\sum_{i=1}^{r(a_d, b_d) - 1} & \\ U(d - 1, 0, (2^{-i}(b_d - a_d), \ldots, 2^{-i}(b_d - a_d))) & \\ \quad + (d - 1) + r(a_d, b_d), & d > 1 \text{ and } \|b - a\| > 2\varepsilon. \end{cases}$$

We observe the following important facts about the function U:

- U is translation invariant, i.e., for every $v \in \mathbb{R}^d$, $U(d, a, b) = U(d, a + v, b + v)$. This is evident from the fact that U depends only on d, ε, and the components of $b - a$.
- U is positive. We see that $r(\cdot, \cdot) > 0$, $d > 0$, and $d - 1 > 0$ when $d > 1$; the positivity of U follows from a simple inductive argument.
- For fixed d, U is monotonic with respect to set inclusion, i.e., if $a, b, a', b' \in \mathbb{R}^d$ and $D_{a,b} \subseteq D_{a',b'}$, then $U(d, a, b) \leqslant U(d, a', b')$. We prove this by examining the following cases:
  - $d = 1$. Monotonicity follows from the fact that $r(a_d, b_d) \leqslant r(a'_d, b'_d)$.
  - $d > 1$ and $\|b - a\|, \|b' - a'\| \leqslant 2\varepsilon$. Here $U = d$ in both cases.

- $d = 2$, $||b - a|| \leqslant 2\varepsilon$, and $||b' - a'|| > 2\varepsilon$. Here $U(d, a, b) = 2 \leqslant (d-1) + r(a'_d, b'_d) \leqslant U(d, a', b')$.
- $d > 2$, $||b - a|| \leqslant 2\varepsilon$, and $||b' - a'|| > 2\varepsilon$. We assume inductively that for all $u, v, u', v' \in \mathbb{R}^{d-1}$ such that $D_{u,v} \subseteq D_{u',v'}$, $||v - u|| \leqslant 2\varepsilon$, and $||v' - u'|| > 2\varepsilon$, we have $U(d-1, u, v) \leqslant U(d-1, u', v')$ (this holds when $d = 3$). By this assumption $U(d-1, 0, \Lambda_S(b' - a')) \geqslant d - 1$; since $r(a'_d, b'_d) \geqslant 1$, it follows that $U(d, a', b') \geqslant d = U(d, a, b)$.
- $d > 1$ and $||b - a||, ||b' - a'|| > 2\varepsilon$. We assume inductively that for all $u, v, u', v' \in \mathbb{R}^{d-1}$ such that $D_{u,v} \subseteq D_{u',v'}$ and $||v - u||, ||v' - u'|| > 2\varepsilon$, we have $U(d-1, u, v) \leqslant U(d-1, u', v')$ (this holds when $d = 2$). As a result of the previous cases, this is equivalent to assuming that U is monotonic for dimension $d - 1$. By this assumption, $U(d-1, 0, \Lambda_S(b - a)) \leqslant U(d-1, 0, \Lambda_S(b' - a'))$ and
$$U(d-1, 0, (2^{-i}(b_d - a_d), \dots, 2^{-i}(b_d - a_d)))$$
$$\leqslant U(d-1, 0, (2^{-i}(b'_d - a'_d), \dots, 2^{-i}(b'_d - a'_d)))$$
for $i = 1, \dots, r(a'_d, b'_d) - 1$. In addition, $r(a_d, b_d) \leqslant r(a'_d, b'_d)$, so U is monotonic for dimension d.

We now show that $N(d, a, b) \leqslant U(d, a, b)$ for all permissible $a, b \in \mathbb{R}^d$, first considering the case $d = 1$. The univariate bisection loop terminates by satisfying the conditions of step 3 or step 6, if by no other way. The analysis of these steps shows that p can be no greater than $\lceil \max(1, \log_2((b - a)/\varepsilon)) \rceil$ at the end of the loop, so $N(1, a, b) \leqslant U(1, a, b)$.

For the case $d > 1$, we assume inductively that $N(d-1, a, b) \leqslant U(d-1, a, b)$ for all permissible $a, b \in \mathbb{R}^{d-1}$. If $||b - a|| \leqslant 2\varepsilon$ then $N(d, a, b) \leqslant d = U(d, a, b)$ due to the action of step 7, so we assume that $||b - a|| > 2\varepsilon$. We now examine the complexity of the various recursive calls.

- Step 8 makes a recursive call on a $d-1$-dimensional function defined on the domain $D_{\Lambda_S(a), \Lambda_S(b)}$. By our inductive assumption and the invariance of U, the number of evaluations performed by the recursive call is bounded above by $U(d-1, 0, \Lambda_S(b - a))$.
- If the loop terminates at step 11 then this step makes a recursive call on a $d-1$-dimensional function defined on the domain $D_{a',b'}$, where $||b' - a'|| \leqslant 2\varepsilon$, as we observed in the analysis of this step. By our inductive assumption, the number of evaluations performed by the recursive call is bounded above by $U(d-1, a', b') = d - 1$.
- Step 12 makes a recursive call on a $d-1$-dimensional function defined on the domain $D_{a',b'}$, where $||b' - a'|| \leqslant 2^{-p}(b_d - a_d)$, as we observed in the analysis of this step. By our inductive assumption and the invariance and monotonicity of U, the number of evaluations performed by the recursive call is bounded above by $U(d-1, 0, (2^{-p}(b_d - a_d), \dots, 2^{-p}(b_d - a_d)))$.

- Step 13 makes a recursive call on a $d-1$-dimensional function defined on the domain D_{a^*,b^*}, where $\|b^*-a^*\| \leqslant 2^{-p}(b_d-a_d)$, as we observed in the analysis of this step. By our inductive assumption and the invariance and monotonicity of U, the number of evaluations performed by the recursive call is bounded above by $U(d-1, 0, (2^{-p}(b_d-a_d), \ldots, 2^{-p}(b_d-a_d)))$.

We define p' as the value of p at the end of the multivariate bisection loop. This loop terminates by satisfying the conditions of step 11 or step 13, if by no other way.

- If the loop terminates at step 11 then $p' \leqslant \lceil \max(1, \log_2((b_d-a_d)/\varepsilon)) \rceil$. If $p' > 1$ then the first $p'-1$ iterations of the loop each made a recursive call in either step 12 or step 13. In addition, step 11 made a recursive call before terminating.
- If the loop terminates at step 13 then $p' \leqslant \lceil \log_2((b_d-a_d)/\varepsilon) \rceil - 1$. The p' iterations of the loop each made a recursive call in either step 12 or step 13.

We see that the worst case occurs when $p' = \lceil \max(1, \log_2((b_d-a_d)/\varepsilon)) \rceil = r(a_d, b_d)$ and the loop terminates at step 11, so we make these assumptions. Combining the recursive call of step 8, the $p'-1$ recursive calls in step 12 or step 13, the final recursive call in step 11, and the p' evaluations of f_d, we obtain

$$\begin{aligned} N(d,a,b) \leqslant\ & U(d-1, 0, \Lambda_S(b-a)) \\ & + \sum_{i=1}^{r(a_d,b_d)-1} U(d-1, 0, (2^{-i}(b_d-a_d), \ldots, 2^{-i}(b_d-a_d))) \\ & + (d-1) + r(a_d, b_d) = U(d,a,b). \end{aligned}$$

Having shown that $N \leqslant U$ in general, we now seek to prove the hypothesis with U in place of N, i.e., $U(d, 0, (1, \ldots, 1)) \leqslant B(d,r)$. We assume that $a = 0$ and $b = (c, \ldots, c)$, where $c > 2\varepsilon$. For such a domain we have $\log_2(c/\varepsilon) > 1$, $p' \geqslant 2$, $2^{-(p'-1)}c \leqslant 2\varepsilon$, and $2^{-i}c > 2\varepsilon$ for $i = 0, \ldots, p'-2$. Thus we obtain

$$U(d, 0, (c, \ldots, c)) = \sum_{i=0}^{p'-2} U(d-1, 0, (2^{-i}c, \ldots, 2^{-i}c)) + 2(d-1) + p'. \tag{13}$$

We define the function

$$M(d,j) \equiv U(d, 0, (2^{-j}, \ldots, 2^{-j}))$$

for $d \geqslant 1$ and nonnegative integers j satisfying $2^{-j} > 2\varepsilon$. The fact that $\varepsilon < 0.5$ (hence $r \geqslant 2$) implies that $0, \ldots, r-2$ are the permissible values of j. We observe that $M(d,0) = U(d, 0, (1, \ldots, 1))$, so we seek to prove that $M(d,0) \leqslant B(d,r)$.

From the definition of U we obtain $M(1,j) = r - j = p'$, so that $M(1,0) = r < r + 1 = C(1,r) - C(0,r) + 2C(0,r+2)$. When $d > 1$ we obtain from (13)

$$\begin{aligned} M(d,j) &= \sum_{i=0}^{r-j-2} M(d-1,i+j) + 2(d-1) + r - j \\ &= \sum_{i=2}^{r-j} M(d-1,r-i) + 2(d-1) + r - j. \end{aligned} \tag{14}$$

We present a simpler expression for (14):

$$M(d,j) = \sigma(d,2,r-j) + 2\sum_{i=0}^{d-1}(d-i)\sigma(i,2,r-j) - 1. \tag{15}$$

We prove (15) using induction. We substitute $d = 2$ into (15) and the definition of σ to obtain

$$\begin{aligned} M(2,j) &= \sigma(2,2,r-j) + 2\sigma(1,2,r-j) + 3 \\ &= \sum_{i=2}^{r-j}(i-1) + 2(r-j-1) + 3 = \sum_{i=2}^{r-j} i + 2 + r - j \\ &= \sum_{i=2}^{r-j}(r-(r-i)) + 2 + r - j = \sum_{i=2}^{r-j} M(1,r-i) + 2 + r - j, \end{aligned}$$

which agrees with (14). We now assume that (15) holds for some $d \geqslant 2$ and all permissible j. From (14) we obtain

$$\begin{aligned} M(d+1,j) &= \sum_{i=2}^{r-j} M(d,r-i) + 2d + r - j \\ &= \sum_{i=2}^{r-j}\left(\sigma(d,2,i) + 2\sum_{k=0}^{d-1}(d-k)\sigma(k,2,i) - 1\right) + 2d + r - j \\ &= \sigma(d+1,2,r-j) + 2\sum_{i=0}^{d-1}(d-i)\sigma(i+1,2,r-j) - (r-j-1) \\ &\quad + 2d + r - j \\ &= \sigma(d+1,2,r-j) + 2\sum_{i=1}^{d}(d+1-i)\sigma(i,2,r-j) + 2(d+1) - 1 \\ &= \sigma(d+1,2,r-j) + 2\sum_{i=0}^{d}(d+1-i)\sigma(i,2,r-j) - 1. \end{aligned}$$

This proves (15).

We set $j = 0$ in (15) to obtain

$$M(d,0) = \sigma(d,2,r) + 2\sum_{i=0}^{d-1}(d-i)\sigma(i,2,r) - 1. \tag{16}$$

We use a simple sum expansion and Lemma A.1 to obtain

$$\sum_{i=0}^{d-1}(d-i)\sigma(i,2,r) = \sum_{i=0}^{d-1}\sum_{k=0}^{i}\sigma(k,2,r)$$
$$= \sum_{i=0}^{d-1}\sigma(i,2,r+1) = \sigma(d-1,2,r+2). \tag{17}$$

Substituting (17) into (16) and applying Lemmas A.3 and A.4, we obtain

$$\begin{aligned} M(d,0) &= \sigma(d,2,r) + 2\sigma(d-1,2,r+2) - 1 \\ &= \sigma(d,1,r) - \sigma(d-1,1,r) + 2(\sigma(d-1,1,r+2) \\ &\quad - \sigma(d-2,1,r+2)) - 1 \\ &= C(d,r) - C(d-1,r) + 2(C(d-1,r+2) - C(d-2,r+2)) - 1 \\ &< B(d,r). \qquad \square \end{aligned}$$

Appendix C. Lipschitz continuity of f^1

We show that the function $f^1 : [0,1]^d \to [0,1]^d$ $(d>0)$ is Lipschitz continuous with constant 1 with respect to the infinity norm, where f^1 is defined for $i = 1, \dots, d$ as

$$f_i^1(x) \equiv \begin{cases} 0.1 + \frac{1}{3}\ln((x_i+1)(x_{(i \bmod (d-1))+1}+1)(x_{((i+1) \bmod (d-1))+1}+1)), & i \text{ even}, \\ 0.4 + \frac{1}{3}\sin(x_i + x_{(i \bmod (d-1))+1} + x_{((i+1) \bmod (d-1))+1}), & i \text{ odd}. \end{cases}$$

We choose arbitrary even $i \in I_d$ and arbitrary $u, v \in [0,1]^d$ to obtain (using the fact that $\ln(1+x) \leqslant x$ when $x \geqslant 0$)

$$\begin{aligned} |f_i^1(u) - f_i^1(v)| &= \frac{1}{3}|\ln((u_i+1)(u_{(i \bmod (d-1))+1}+1)(u_{((i+1) \bmod (d-1))+1}+1)) \\ &\quad - \ln((v_i+1)(v_{(i \bmod (d-1))+1}+1)(v_{((i+1) \bmod (d-1))+1}+1))| \\ &= \frac{1}{3}\left|\ln\frac{u_i+1}{v_i+1} + \ln\frac{u_{(i \bmod (d-1))+1}+1}{v_{(i \bmod (d-1))+1}+1} + \ln\frac{u_{((i+1) \bmod (d-1))+1}+1}{v_{((i+1) \bmod (d-1))+1}+1}\right| \\ &\leqslant \frac{1}{3}\left(\frac{|u_i - v_i|}{v_i+1} + \frac{|u_{(i \bmod (d-1))+1} - v_{(i \bmod (d-1))+1}|}{v_{(i \bmod (d-1))+1}+1}\right. \\ &\quad \left. + \frac{|u_{((i+1) \bmod (d-1))+1} - v_{((i+1) \bmod (d-1))+1}|}{v_{((i+1) \bmod (d-1))+1}+1}\right) \leqslant ||u-v||. \end{aligned}$$

We choose arbitrary odd $i \in I_d$ and arbitrary $u, v \in [0,1]^d$, set

$$u' = u_i + u_{(i \bmod (d-1))+1} + u_{((i+1) \bmod (d-1))+1},$$
$$v' = v_i + v_{(i \bmod (d-1))+1} + v_{((i+1) \bmod (d-1))+1},$$

and assume w.l.o.g. that $\sin(u') \geqslant \sin(v')$. We obtain

$$|f_i^1(u) - f_i^1(v)| = \frac{1}{3}(\sin(u') - \sin(v')) = \tfrac{1}{3}\int_{v'}^{u'} \cos(\zeta)\, d\zeta \leqslant \frac{1}{3}\left|\int_{v'}^{u'} d\zeta\right| = \frac{1}{3}|u' - v'| \leqslant ||u - v||.$$

Appendix D. Pseudocode

We summarize the algorithm description of Section 5.1 in the following pseudocode. Given a dimension $d > 0$ and a function $f \in \mathfrak{F}^{\varepsilon}_{a,b}$, the function call $\mathbf{fp}(f, d, a, b, \varepsilon)$ returns an ε-residual solution for f, where the function **fp** is defined below.

$\mathbf{fp}(f, d, a, b, \varepsilon)$
! **fp** returns a point $x \in D_{a,b}$ satisfying $||f(x) - x|| \leqslant \varepsilon$.
! f is a d-dimensional function in $\mathfrak{F}^{\varepsilon}_{a,b}$.
! $d > 0$ is the number of variables (dimensions) of f.
! a,b are real d-vectors such that $a_i \leqslant b_i$ for $i = 1, \ldots, d$.
! $\varepsilon > 0$ is the error tolerance.
! **Step 1**
if $d = 1$ then
  $x^- := a$;
  $x^+ := b$;
  if $x^- = x^+$ then
    return x^-;
  end(if)
  repeat
! **Step 2**
    $x := (x^+ + x^-)/2$;
    $u := f(x)$;
    if $|u - x| \leqslant \varepsilon$ then
      return x;
    end(if)
! **Step 3**
    if $x^- = a$ and $x - x^- \leqslant \varepsilon$ and $u < x$ then
      return x^-;
    end(if)
    if $x^+ = b$ and $x^+ - x \leqslant \varepsilon$ and $u > x$ then
      return x^+;
    end(if)

```
  ! Step 4
        if $u > x$ then
              $x^- := \min(x^+, (x+u)/2)$;
        else
              $x^+ := \max(x^-, (x+u)/2)$;
        end(if)
        if $x^- = x^+$ then
              return $x^-$;
        end(if)
! Step 5
        if $x^- = a$ and $x^+ - x^- \leqslant \varepsilon/2$ and $u < x$ then
              return $a$;
        end(if)
        if $x^+ = b$ and $x^+ - x^- \leqslant \varepsilon/2$ and $u > x$ then
              return $b$;
        end(if)
! Step 6
        if $x^- \neq a_1$ and $x^+ \neq b$ and $x^+ - x^- \leqslant \varepsilon$ then
              return $(x^+ + x^-)/2$;
        end(if)
  end(repeat)
else
! Step 7
  if $||b - a|| \leqslant 2\varepsilon$ then
        $x := ((a_1 + b_1)/2, \ldots, (a_d + b_d)/2)$;
        for $i = 1$ to $d$ do
              if $a_i = b_i$ then
                    $\tilde{x}_i := a_i$;
              else
                    $\tilde{x}_i := \max(a_i, \min(b_i, f_i(x)))$;
              end(if)
        end(for)
        return $\tilde{x}$;
  end(if)
! Step 8
  $S := 1, \ldots, d-1$;
  $g(\cdot) := \Lambda_S(f(\Psi_S(\cdot, (a_d + b_d)/2)))$ defined on the domain $D_{\Lambda_S(a), \Lambda_S(b)}$;
  $x := \Psi_S(\mathbf{fp}(g, d-1, \Lambda_S(a), \Lambda_S(b), \varepsilon), (a_d + b_d)/2)$;
  $x^- := x[d \leftarrow a_d]$;
  $x^+ := x[d \leftarrow b_d]$;
  if $x_d^- = x_d^+$ then
        return $x$;
  end(if)
  repeat
```

```
! Step 9
        $u_d := f_d(x)$;
        if $|u_d - x_d| \leqslant \varepsilon$ then
                return $x$;
        end(if)
! Step 10
        if $u_d > x_d$ then
                $x^- := x$;
        else
                $x^+ := x$;
        end(if)
! Step 11
        if $x_d^- = a_d$ and $x_d^+ - x_d^- \leqslant \varepsilon$ then
                $y := x_d^-$;
                for $i := 1$ to $d-1$
                        $a_i' := \max(a_i, x_i^+ - ||y - x_d^+||)$;
                        $b_i' := \min(b_i, x_i^+ + ||y - x_d^+||)$;
                end(for)
                $g(\cdot) := \Lambda_S(f(\Psi_S(\cdot, y)))$ defined on the domain $D_{a',b'}$;
                return $\Psi_S(\mathbf{fp}(g, d-1, a', b', \varepsilon), y)$;
        end(if)
        if $x_d^+ = b_d$ and $x_d^+ - x_d^- \leqslant \varepsilon$ then
                $y := x_d^+$;
                for $i := 1$ to $d-1$
                        $a_i' := \max(a_i, x_i^- - ||y - x_d^-||)$;
                        $b_i' := \min(b_i, x_i^- + ||y - x_d^-||)$;
                end(for)
                $g(\cdot) := \Lambda_S(f(\Psi_S(\cdot, y)))$ defined on the domain $D_{a',b'}$;
                return $\Psi_S(\mathbf{fp}(g, d-1, a', b', \varepsilon), y)$;
        end(if)
! Step 12
        if $x_d^- = a_d$ then
                $y := (x_d^+ + x_d^-)/2$;
                for $i := 1$ to $d-1$
                        $a_i' := \max(a_i, x_i^+ - ||y - x_d^+||)$;
                        $b_i' := \min(b_i, x_i^+ + ||y - x_d^+||)$;
                end(for)
                $g(\cdot) := \Lambda_S(f(\Psi_S(\cdot, y)))$ defined on the domain $D_{a',b'}$;
                $x := \Psi_S(\mathbf{fp}(g, d-1, a', b', \varepsilon), y)$;
        else if $x_d^+ = b_d$ then
                $y := (x_d^+ + x_d^-)/2$;
                for $i := 1$ to $d-1$
                        $a_i' := \max(a_i, x_i^- - ||y - x_d^-||)$;
```

$b'_i := \min(b_i, x_i^- + ||y - x_d^-||)$;
end(for)
$g(\cdot) := \Lambda_S(f(\Psi_S(\cdot, y)))$ defined on the domain $D_{a',b'}$;
$x := \Psi_S(\mathbf{fp}(g, d-1, a', b', \varepsilon), y)$;
else
! **Step 13**
$y := (x_d^+ + x_d^-)/2$;
for $i := 1$ to $d - 1$
$a'_i := \max(a_i, x_i^- - ||y - x_d^-||, x_i^+ - ||y - x_d^+||)$;
$b'_i := \min(b_i, x_i^- + ||y - x_d^-||, x_i^+ + ||y - x_d^+||)$;
end(for)
$g(\cdot) := \Lambda_S(f(\Psi_S(\cdot, y)))$ defined on the domain $D_{a',b'}$;
$x := \Psi_S(\mathbf{fp}(g, d-1, a', b', \varepsilon), y)$;
if $x_d^+ - x_d^- \leqslant 2\varepsilon$ then
return x;
end(if)
end(if)
end(repeat)
end(if)

References

[1] G. Allgower, K. Georg, Numerical Continuation Methods, Springer, New York, Berlin, 1990.
[2] S. Banach, Sur les operationes dans les ensembles abstracts et leur application aux equationes integrales, Fund. Math. 3 (1922) 133–181.
[3] G. Birkhoff, Trans. Amer. Math. Soc. 18 (1917) 199.
[4] T. Bountis, R. Helleman, J. Math. Phys. 22 (1981) 1867.
[5] B.C. Eaves, Homotopies for computation of fixed points, Math. Programming 3 (1972) 1–22.
[6] B.C. Eaves, R. Saigal, Homotopies for computation of fixed points on unbounded regions, Math. Programming 3 (1972) 225–237.
[7] J. Greene, A method for determining a stochastic transition, J. Math. Phys. 20 (1979) 1183–1201.
[8] R. Helleman, in: U. Landman (Ed.), Statistical Mechanics and Statistical Methods in Theory and Application: On the Iterative Solution of a Stochastic Mapping, Plenum, New York, 1977, p. 343.
[9] M. Henon, Numerical study of quadratic area-preserving mappings, Quart. Appl. Math. 27 (1969) 291–312.
[10] M.D. Hirsch, C. Papadimitriou, S. Vavasis, Exponential lower bounds for finding Brouwer fixed points, J. Complexity 5 (1989) 379–416.
[11] J. Howland, R. Vaillancourt, Attractive cycles in the iteration of meromorphic functions, Numer. Math. 46 (1985) 323–337.
[12] Z. Huang, L. Khachiyan, K. Sikorski, Approximating fixed points of weakly contracting mappings, J. Complexity 15 (1999) 200–213.
[13] A. Nemirovsky, private communication, 1999.
[14] A. Nemirovsky, D.B. Yudin, Problem Complexity and Method Efficiency in Optimization, Wiley, New York, 1983.
[15] M. Scarf, The approximation of fixed points of a continuous mapping, SIAM J. Appl. Math. 15 (1967) 1328–1343.

[16] S. Shellman, K. Sikorski, Algorithm 825: A deep-cut bisection envelope algorithm for fixed points, ACM TOMS, Vol. 29, No. 3, Sept. 2003, pp. 309–325.
[17] S. Shellman, K. Sikorski, A two-dimensional bisection envelope algorithm for fixed points, J. Complexity 18 (2002) 641–659.
[18] K. Sikorski, Optimal Solution of Nonlinear Equations, Oxford Press, New York, 2001.
[19] K. Sikorski, C.W. Tsay, H. Woźniakowski, An ellipsoid algorithm for the computation of fixed points, J. Complexity 9 (1993) 181–200.
[20] F. Verhulst, Nonlinear Differential Equations and Dynamical Systems, Springer, Berlin, 1990.
[21] M. Vrahatis, An efficient method for locating and computing periodic orbits of nonlinear mappings, J. Comput. Phys. 119 (1995) 105–119.
[22] Z. Yang, Computing Equilibria and Fixed Points, Kluwer Academic Publishers, Boston, Dordrecht, London, 1999.